The Dissertation Committee for Hyun Jin Kim
certifies that this is the approved version of the following dissertation:

BIST Methodology for Low-Cost Parametric Timing Measurement of High-Speed Source Synchronous Interfaces

Committee:

Jacob A. Abraham, Supervisor

Ranjit Gharpurey

David Z. Pan

Michael Orshansky

John X. J. Zhang

BIST Methodology for Low-Cost Parametric Timing Measurement of High-Speed Source Synchronous Interfaces

by

Hyun Jin Kim, B.S.E.; M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
May 2012

Dedicated to my parents.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Jacob A. Abraham. He granted me to work in his research group and gave me countless insightful comments and guidance. He has always given me the support and encouragement to finish my Ph.D. study. I would like to thank Dr. David Z. Pan, Dr. Ranjit Gharpurey, Dr. Michael Orshansky and Dr. John X. J. Zhang for being my committee and providing insightful feedback on my research.

My gratitude goes to my friends in the Computer Engineering Research Center, Jae-Wook Lee, Joonsung Park, Jihwan Chun, Jaeyong Chung, Kihyuk Han, Eun Jung Jang, Junyoung Park, Sriram Sambamurthy, Mahesh Prabhu, Hsun-Cheng Lee, Pratyusha Nidamaluri, Shahrzad Mirkhani, Ameya Chaudhari, Chaoming Zhang, Shih-Hsin Jason Hu. I would like to appreciate friends at University of Texas at Austin (UT), Jae Hong Min, Hyejeong Song, Jiwoo Pak, Sangman Kim, Lindsay Kowis, Nisha Ganwani, Anurag Kumar, Jaeseok Yang, Yongchan Ban, Kayoung Lee, Heejung Park, Katrina Lu, Kun Yuan, Ashutosh Chakraborty. Also, I would like to acknowledge Melissa Campos, Debi Prather and Melanie Gulick for administrative support.

I wish to thank my friends Eunmi Chu, Kisun Shin, Mihwa Choi, Sunghua Hong, Hyoungsik Nam, Jeongpyo Kim, Chan Hong Park, Hong Chul

Kim, Hyong Ki Ahn, Hyunseok Kim, Jae Joon Kim, Jeongsik Yang, Jinwook Kim, Sangjin Byun, Taesung Kim, Yongchul Song.

I would like to thank my co-workers at Samsung Electronics, Okjoo Park, Sunmi Lim, Mijin Lee, Semi Yang, Sujin Yang, Nayoung Kim, Hyeran Kim, Sunmi Kim, Sangjoon Hwang, Kwangil Park, Daehyun Kim, Jeongdon Lim, Chelwoo Park, Haksu Yu, Hyundong Kim, Younsik Park, Sungjin Jang, Youngwook Jang, Seokwon Hwang, Joosun Choi. Special thanks to Younghyun Jun, who was my manager at Samsung and gave me the opportunity to start Ph.D study.

Finally, I am grateful to my parents for their patience, love and support. Without them this work would never have come into existence.

BIST Methodology for Low-Cost Parametric Timing Measurement of High-Speed Source Synchronous Interfaces

Hyun Jin Kim, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Jacob A. Abraham

With the scaling of technology nodes, the speed performance of microprocessors has rapidly improved but the scaling of off-chip input/output (I/O) bandwidth is limited by physical pin resources and interconnect technologies. In order to reduce the performance gaps, new interface techniques have emerged and the marketplace has moved towards higher levels of integration with system on a chip (SoC) implementations. The advent of new techniques, however, has led to new challenges on the semiconductor and automated test equipment (ATE) industries. The relatively slow growing ATE technology comparing to I/O speeds especially intensifies manufacturing test issues. Testing high speed I/O timing parameters requires expensive high performance test equipment with high accuracy and resolution. The requirements increase integrated circuit (IC) manufacturing costs and thus test issues have become critical.

This thesis focuses on on-chip test methods to improve test accuracy and reduce test costs for high speed double data rate (DDR) memory I/Os using source synchronous clocking. For testing the I/O timing parameters, a phase interpolator based on-chip timing sampler using a cycle-by-cycle control method was developed. This circuit generates data and clock patterns and controls the time delay between data and clock to detect the timing mismatch which indicates timing degradations. The on-chip timing sampler was implemented as a built-in self test (BIST) circuit for low-cost parametric timing measurements. The BIST scheme was fabricated with a 0.18-μm CMOS process technology. Using the static and dynamic modes, measurement results are obtained for the I/O timing parameters such as the setup and hold times, input voltage-level variations tolerances, duty distortion tolerances and data skews. Moreover, a delay mismatch measurement method was developed to improve measurement accuracy using a simple control circuit. This delay mismatch detector measures timing mismatches between data and clock paths and then the timing mismatches are converted to timing specifications. This scheme is also implemented along with analog to digital converter (ADC) to collect digital test results supporting low-cost system-level tests. Thus, the low-frequency test results show that our on-chip measurement techniques provide an attractive low-cost solution and are effectively applied for testing high speed source synchronous systems.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the scaling of technology nodes, the performance gap between processor and memory has rapidly increased. As described in [43], while the main target of processor fabrication lines is to make fast logics, memory fabrication lines focus on the characteristic of low leakage current. Moreover, the separation of processor and memory manufacturers is causing a growing disparity in performance. While microprocessor performance has been improving at a rate of 60% per year, the access time to dynamic random access memory (DRAM) has been improving at less than 10% per year [50]. In order to overcome the performance disparity, memory devices need to speed up the memory clock or increase the bus width. However, speeding up the clock frequency requires stringent timing constraints for systems. Increasing bus width can be done at the expense of the pin count and I/O power [43]. In order to decrease the requirements of large pin counts, memory interfaces need to operate at higher data-rates. Thus, memory devices such as rambus dynamic random access memory (RDRAM) [24, 30] and double data rate (DDR) synchronous DRAM (SDRAM) [1] have been developed to improve the interface bandwidth. In 2009 ITRS [63], the high speed interface trend in computing and network applications was presented as shown in Figure 1.1. The per pin data rate of DDR memory has been increasing rapidly and is expected to be over 6 Gigabit per second (Gbps) by 2020. The increasing bandwidth brings new challenges in design and testing industries. Especially, testing issues have been increasingly critical because testing technologies do not grow as rapidly as design technologies. Most previous work on memory test targeted testing the core memory operations by using fault models as proposed in [53]. On the other hand, this dissertation describes test techniques related to the less studied area of memory interface timing to obtain low-cost parametric timing measurement results.

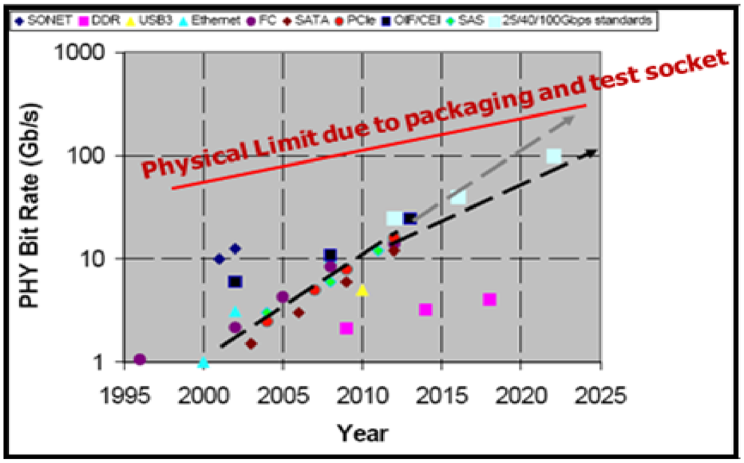


Figure 1.1: High speed interface trend in computing and network applications

1.1 Background

This dissertation describes a source synchronous DDR memory interface as an example for BIST applications. In order to tackle testing issues of high speed source synchronous interfaces, we first introduce the basic structure of source synchronous DDR interfaces. We then provide recent test architecture trends to discuss motivations for our work.

1.1.1 High Speed Source Synchronous DDR Interface

As the high-bandwidth requirements of computer systems increase, source synchronous parallel links using double data rate (DDR) signalling (data is sent at both rising and falling edges of clock) have been developed. The source synchronous DDR interfaces eliminate a clock recovery circuit and achieve twice the I/O bandwidth of a single data rate. Figure 1.2 shows the source synchronous interface structure. The data and the dedicated strobe are transferred together from a transmitter (TX) to a receiver (RX), and the RX captures the data using the sent strobe. This interface technique is devised to reduce timing jitter between data and clock/strobe by sourcing the strobe along with the data. A representative source synchronous device is

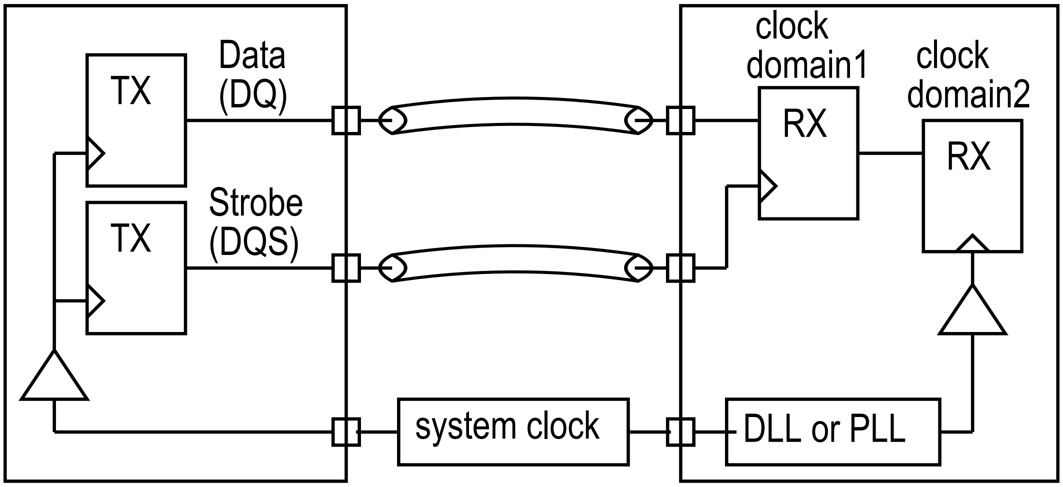


Figure 1.2: Source-synchronous interface structure

DDR SDRAM [23], and thus our work focuses on test issues for this memory architecture.

1.1.2 Testing Issues of High Speed Memory Interfaces

According to the 2009 ITRS Roadmap [63] shown in Figure 1.3, the per pin data rate of memory has been increasing rapidly and is expected to be over 6 Gbps by 2020. DDR2 [1], DDR3 [2] and DDR4 [23] interfaces have been developed to satisfy the increasing data rate requirements of memory. The trend in data rates has made it challenging to measure memory interface timing parameters with high accuracy. As the memory speed has reached over 1 Gigabit per second (Gbps), any small timing mismatches between data and strobe need to be considered to satisfy timing specifications [10]. Because of the requirements of high resolution and high accuracy, testing the high-speed DDR

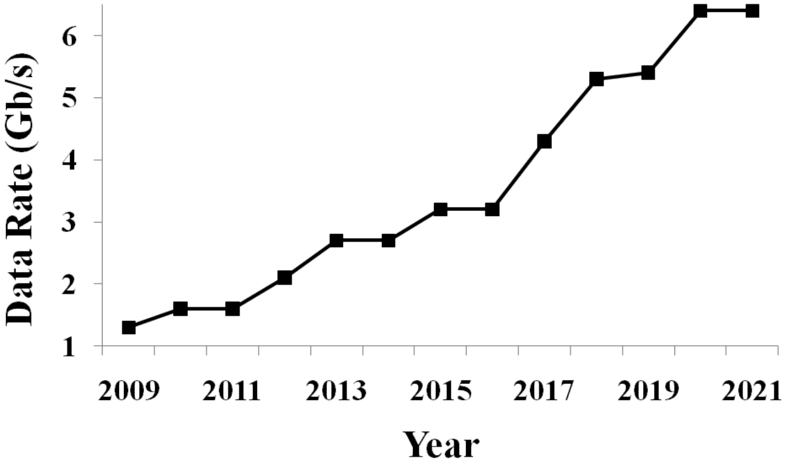


Figure 1.3: Memory I/O data rate [2009 ITRS]

interface timings has become challenging. Moreover, a conventional automated test equipment (ATE) has limitations for testing source synchronous interfaces.

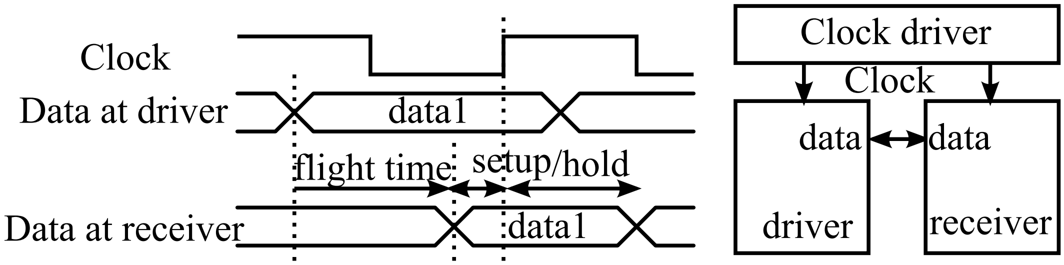
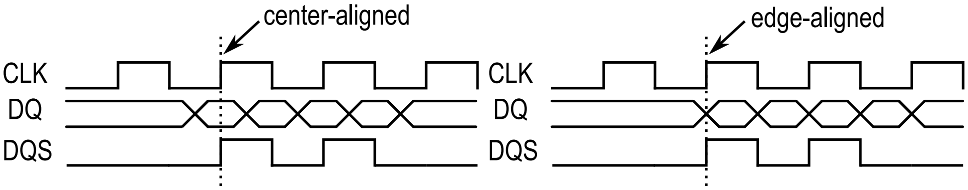


Figure 1.4: Global clock I/O timing and block diagram

The conventional ATE has been implemented to test global clock interfaces, which are available for below 100 Mbps data rate. Figure 1.4 shows the structure and timing for the global clocking scheme where a system clock is used for driving and sampling data both at the driver and the receiver. Testers for the global clocking interfaces do not need to synchronize data between device under test (DUT) and tester. Thus, the conventional ATE simply provides drivers and comparators. On the other hand, for testing source synchronous interfaces, a strobe-clock needs to be implemented on ATE to synchronize data between DUT and ATE.

To help understandings in the requirements for testing source synchronous interfaces, read and write interface timings for memory devices are illustrated in Figure 1.5 where CLK, DQ and DQS indicate an external clock, data and data-strobe, respectively. As can be seen in Figure 1.5, during a write to the memory device, the DQS and CLK must be center-aligned with respect to DQ. On the other hand, the DQS and CLK are edge-aligned with respect to



(a) Memory write timing (b) Memory read timing

Figure 1.5: Memory write and read interface timing

DQ for the memory read operation. Therefore, for testing the memory write timing, ATE needs to generate different phases for DQ and DQS/CLK. For the phase alignments, the clock used to capture DQ is a 90 degree phase-offset with respect to the clock used to capture DQS/CLK. For testing the memory read timing, the requirements of ATE are more challenging because of the strobe-clock implementation with high resolution and accuracy. Moreover, the strobe of ATE needs to capture and synchronize the non-deterministic DQ and DQS timings coming from the DUT. As the memory speed increases, uncertainties aggravate the non-deterministic data problem due to:

- Clock/data jitter
- Power supply noise
- Channel noise: inter-symbol interference (ISI)
- Crosstalk
- Device and layout mismatches: random process variations or systematic layout mismatches.

These issues increase the requirements of high edge placement accuracy (EPA) and high resolution for testers [11]. Therefore, test costs and development time for high performance ATE drastically increase. In the next section, we discuss the recent trends of test architectures to deal with these issues.

1.1.3 Test Architecture Trends

Traditionally, as shown in Figure 1.6, the basic test structure consists of ATE, DUT and an electrical connector (probe card). Testing at multi-gigabit per second (multiGbps) requires high performance ATE and also high transfer data rates between ATE and DUT [3, 21, 64]. Thus, the interfacing channels should have a good signal integrity, which increases test costs. The test cost has become a large percentage of the total production cost, and thus a new test

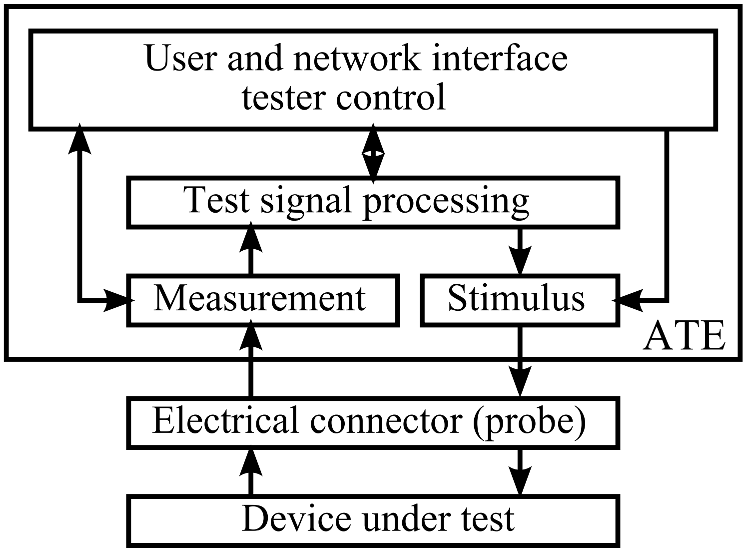


Figure 1.6: Conventional test structure

paradigm is needed to obtain both high quality and low cost [4, 40, 66]. Moreover, the speed and improvement of ATE can not catch up with the increasing speed of device. Therefore, recently, design-for-test (DFT) or built-in-self test (BIST) methods have been developed to overcome a bandwidth limitation between DUT and ATE [28, 57]. As shown in Figure 1.7, the DFT/BIST modules are combined with the existing low-cost testers, hence overall test costs decrease. The DFT/BIST modules can be implemented as a built-off chip or on-chip method. The crucial goal is to reduce test time and cost while the test quality remains.

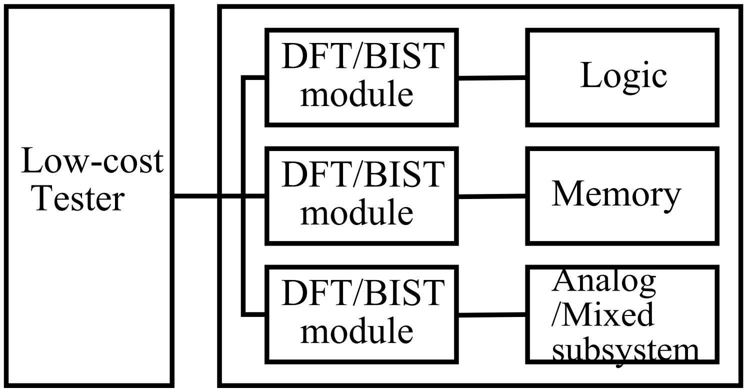


Figure 1.7: Tester combined with DFT/BIST

1.1.4 Motivations for On-Chip Timing Test

The built-off test circuits interfacing between ATE and DUT improve test quality, and reduce a development time and test costs for ATE. On the other hand, off-chip test methods still have constraints for testing high-speed signals as follows:

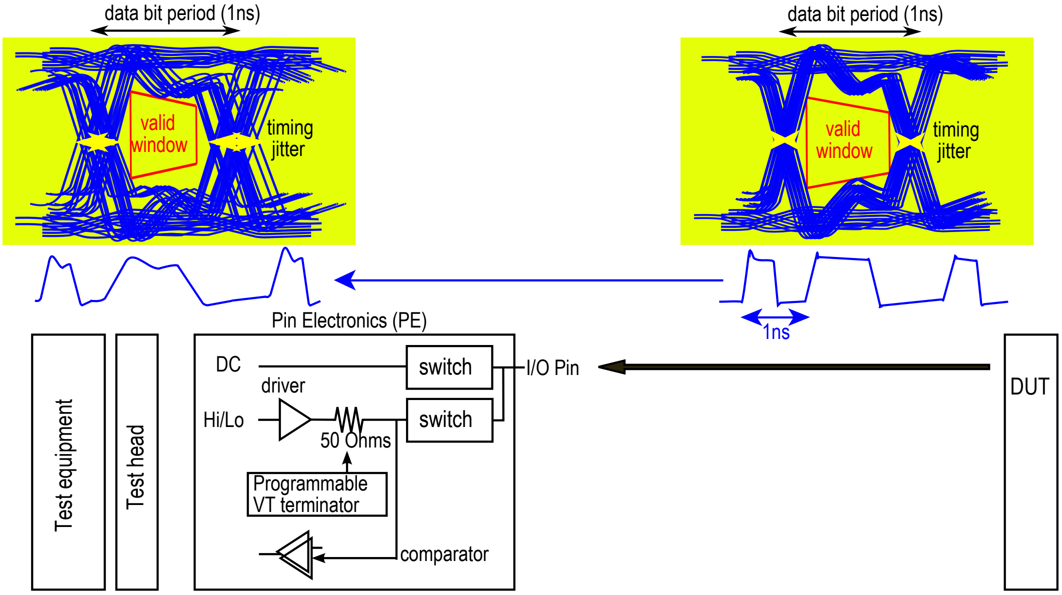


Figure 1.8: Issues for testing high-speed I/O signals over 1 Gbps

- Measurement error due to signal integrity issue
- Timing jitter of the off-chip strobe signal
- AC I/O pin level distortion
- Test time and cost issues for high volume production
- Time-to-market issue

Figure 1.8 shows an example explaining the issues of off-chip test methods for testing high-speed I/O signals over 1 Gbps. A typical test-setup consists of test equipment (for measurement, stimulus and test signal processing), test head, pin electronics card and DUT. The device signals are transmitted through the tester channels from the DUT I/O pins to the tester. The tester measures the timing performance of the signals. During the signal transfer, signal degradations such as duty-ratio distortion, slew-rate change and amplitude distortion occur due to the following signal integrity issues:

- Inter-symbol interference
- Channel impedance mismatches
- Channel loading mismatches
- Board trace length mismatches among parallel I/O pins
- Crosstalk among parallel I/O pins
- Heavy capacitance loadings because of relay-switching - components for an alternative connection used in probe card

For source-synchronous DDR memory systems operating at over 1 Gbps, uncertainties due to the off-chip test environments increase timing jitter and thus significantly increase the measurement error. As can be seen in Figure 1.8, as the data rate increases and the data bit period decreases, the portion of timing jitter increases and thus the valid window increasingly decreases. For the worse case, if the data bit period is 0.5 ns, the eye opening is reduced and the valid window is narrower. For this case, it is very challenging to find the valid window period of the device signals using the strobe of ATE. Thus, timing jitter due to ATE should be minimized, which is very hard. To overcome these issues, calibration techniques are implemented and an expensive

Membrane probe card ($40 ~ $80 per pin) has been used [22, 47]. However, these resources drastically increase test time and costs.

Additionally, AC I/O pin level distortions affect input timing parameters for memory devices. Figure 1.9 shows an example of V_{IL} and V_{IH} input-level distortions, where V_{IL} and V_{IH} indicate input pin levels [26]. Interface circuits in memory devices are designed and optimized to meet the AC I/O input-level specifications which are defined in JEDEC [23]. However, it is challenging for ATE to transmit the test signals with the exact AC input-level to the DUT. Due to the heavy capacitive loadings, a signal swing is degraded and a duty-cycle distortion occurs. Thus, the signal distortions of test inputs significantly degrade the setup and hold times for memory devices. To avoid test inaccuracy and provide an optimum pin level to the DUT, the ATE should calibrate the I/O pin levels. These additional test flows increase test time and costs to validate the timing between data and data strobe.

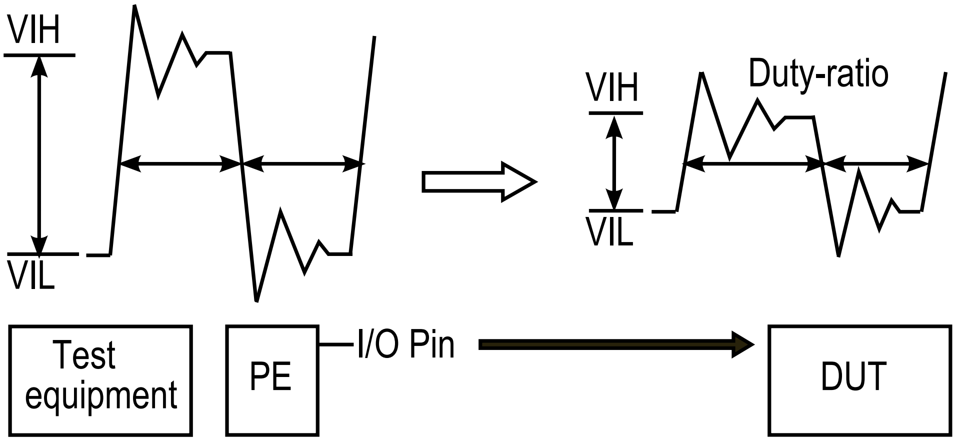


Figure 1.9: I/O pin level distortions (from ATE to DUT)

The most critical issue is that the uncertainties of testers are not the same as the real operations in computer systems. In reality, memory controllers still calibrate the memory device signals to optimize the performance even though ATE determines the devices as ‘PASS’. We need to improve test quality and reduce measurement errors using high-end ATE to increase yields. However, putting a lot of off-chip resources for testing high-speed signals is not very effective. Therefore, an embedded test method has been a promising solution in terms of overall device costs, test costs and yields. An on-chip test method for testing source synchronous memory I/Os reduces the complexity of ATE for the following reasons:

- A complicated circuit and an expensive probe card for high-speed clock are not required
- Additional strobe-clock generations are not required
- Complicated tester timing is not needed
- The bandwidth demand between ATE and DUT is minimized
- High-speed I/O pins at probe are not required
- Higher levels of parallel test is enabled (shorter test times), which is good for high volume production

Additionally, if testing high-speed I/Os using on-chip test methods can be fully performed at wafer probe, time-to-market can be reduced. Generally, testing at wafer probe is performed to sort a known good die as early as possible, and to achieve high yields at package tests. On-chip test circuits enable all functional tests to be done at wafer probe, and thus we can achieve a higher yield at package test and reduce time-to-market. Furthermore, recently, system on a chip (SoC) designs are scaling and test equipment/tooling costs are increasing. Thus, the silicon area on-chip measurement circuits take up is becoming insignificant. Therefore, on-chip measurement circuits have been preferred. Next, we introduce our on-chip methods for testing high-speed DDR memory I/O timings. An embedded test method reduces the pin-count at probe and increases the multi-site test efficiency, and thus reduces test times. Also, the method minimizes the bandwidth demand between ATE and DUT and thus reduces the complexity of tester. Moreover, it enables at-speed testing without at-speed ATE. In general, testing high-speed mix-signals has been performed at package-level, not at wafer probe [45]. Assembly techniques are not sufficiently accurate, and thus at-speed testing for analog/mixed signals at wafer probe still has many challenges. Figure 1.10 illustrates an IC manufacturing process. Using an on-chip measurement methods, all functional tests could be done at wafer probe, and we can achieve a higher package yield and reduce time-to-market. As the technology nodes scale and SOC designs scale-up, a $1 \sim 2\%$ silicon overhead because of the embedded test circuits is becoming negligible. Because of these benefits, on-chip measurement circuits have been presented.

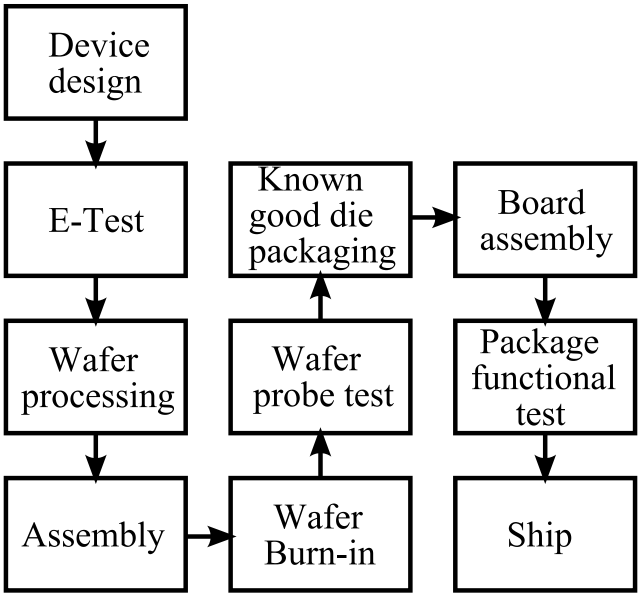


Figure 1.10: IC manufacturing process

1.2 Area of Focus

The discrepancies between the test environment and the realistic system operations cause measurement inaccuracy. Since connections through a 50-ohm coaxial cable between ATE and DUT for testing are required, the additional jitter due to cables causes signal distortions such as amplitude degradation and impedance discontinuities. Moreover, the heavy I/O channel loadings at wafer probe test exacerbate the limitations of ATE. Thus, we have to postpone high speed I/O testing to the package-level, which increases the time-to-market.

To set the context of test challenges, as an example application, this dissertation presents a DDR memory architecture with a representative high-speed parallel interface. Figure 1.11 shows the simplified diagram of DDR SDRAM, which consists of a memory core, input/output circuitry, and a clock generator. The focus of our research has been developing test methodologies for input/output circuitry, which operate at higher data rates.

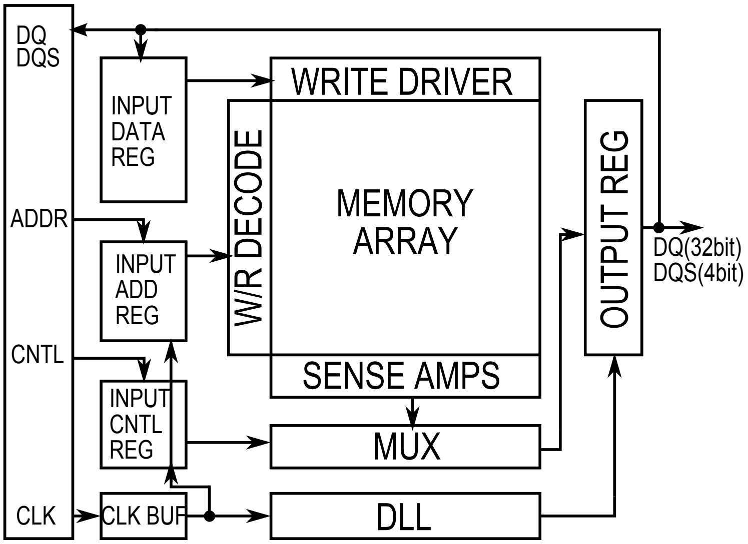


Figure 1.11: Simplified diagram of DDR memory architecture

For testing input/output circuitry, it is important to measure timing margin between transmitter and receiver. The timing margin is the tolerable margin of error and is defined between data and clock signal positions. In the eye diagram of Figure 1.12, the timing margin is ideally equal to the bit period, but is reduced due to various noise sources such as data jitter, clock jitter, and channel noise [37]. As data rates increase, the bit period become smaller but both channel-related and circuit-related noise sources increase. These issues cause ‘close eye’, and make the measurement of timing margin more challenging [5, 6, 17]. On-chip process, voltage, and temperature (PVT) variations

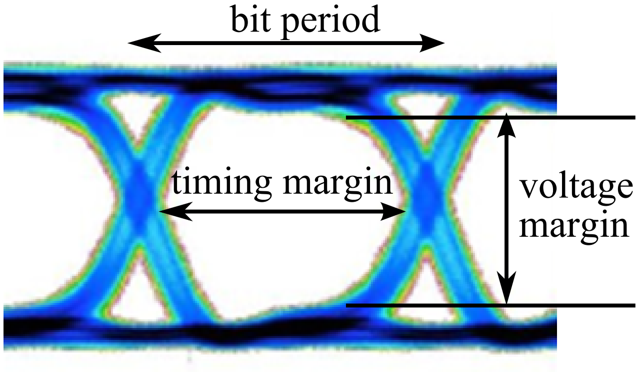


Figure 1.12: Eye diagram

exacerbate test accuracies. The increasing uncertain noise sources degrade test accuracies and thus drop yield and increase time-to-market. Therefore, we need test methods with higher accuracy for high speed parallel I/O timing test.

1.3 Primary Contributions

This dissertation investigates on-chip techniques for testing high-speed source synchronous parallel I/O timing parameters. The main contributions of this work are listed below.

1.3.1 On-Chip Timing Sampler with High Resolution and Low Area Overhead

This dissertation describes two on-chip timing sampler techniques to detect the time delay between data and strobe signals. The one method is a phase interpolator based on-chip timing sampler. This circuit generates the new strobe from the data and strobe signal edges and then controls the edge of the new strobe using a cycle-by-cycle control method. The new strobe for sampling signals keeps shifting at every clock cycle until the time delay is detected. Using this scheme, the test is completed in one test sequence. Since the new strobe is generated with a relative timing relationship between data and strobe, the common jitter is canceled and thus this scheme is little affected by the timing jitter. Another method presented is a delay line based timing sampler. This scheme improves testability and observability because it is a fully digital timing generator. Using a simple delay line, we generate the sampling timing and digitally control the timing delays. Moreover, this scheme implements a dual-capture which enables capturing data at both the rising and the falling edges. These techniques thus supports at-speed testing of DDR interface timings without any limitations of the external clock frequency. Moreover, these timing samplers take small area overhead compared to previous work [13, 70] because these techniques only implement relative time delays and thus need to implement less delay lines.

1.3.2 Low-Cost Measurement Method

We present a low-cost measurement methodology by combining the cycle-by-cycle control method and on-chip timing samplers. Our work does not use any external high-speed signals and the test results are also obtained with digital values or a multiple of the test clock period, and hence our work is a low-cost solution.

This measurement technique is applied for testing the dynamic and static timing margins of the memory interface timings. The dynamic timing margin tests are performed by inserting timing jitter or changing data patterns and thus the dynamic test estimates the worst timing variations. The static timing margin tests are mainly used to obtain timing pass or fail information in a short test time. This test is performed by comparing the timing specifications between the reference and the measurement paths where the reference is determined as shown in JEDEC [23] specs. The test results for the timing margins are a multiple of the test clock period, and we measure the I/O timing parameters simply by counting the number of the test clock cycle. Therefore, our on-chip test methods are compatible with low-cost testers, and thus the applications can be tested using a traditional low-end ATE. These strengths of this work help to reduce overall IC manufacturing costs and also decrease time-to-market issues.

1.3.3 Small Timing Error Detection

With the increasing data rates, the memory interface timing parameters become tighter and thus small timing errors cannot be negligible. Previously, research work related to delay measurement with high resolution has been done [27, 49, 62]. This conventional delay measurement method using delay lines has an offset delay because of the internal capacitive loadings. The offset delay limits the minimum value of the dynamic delay range and thus it has limitations to detect small timing errors of the memory interfaces. The source synchronous data and strobe signals transfer in parallel and thus the timing mismatches between two paths are generally very small (several tens picosecond) in high-speed systems over tens of GHz. In order to detect the small timing errors, we developed a path delay mismatch detector which detects delay mismatches between two paths. The path delay mismatch indicates the timing degradations and is calculated to obtain the I/O timing specifications. A digital assisted design approach is also applied for this scheme to control the timing test resolution.

1.4 Overview of the Dissertation

This thesis is divided into three main parts; the first describes the on-chip timing samplers in Chapter 2 and 4, the second describes a low-cost measurement technique for the memory input and output timing parameters to support a system-level test in Chapter 3 and 5, and the third describes a path delay mismatch detection method to increase testability for higher speed interface timings in Chapter 6. Chapter 7 concludes this thesis with future directions.

In Chapter 2, we present a phase interpolator based on-chip timing sampler to enable the memory interface timing tests without external high-end testers. This scheme generates data and clock patterns, and then generates a new strobe using the timing edges of the generated data and clock signals. Using the new strobe, we detect the timing delay difference between two signals. We also show memory BIST applications using this scheme to measure the I/O timing margins but this scheme is generally applicable for time delay measurements.

Chapter 3 describes a low-cost parametric measurement method for testing memory interface timings. We combined the phase interpolator based on-chip timing sampler and a cycle-by-cycle control method to present the low-cost test scheme. In order to calculate the timing margin of the data with respect to the strobe, we simply need to count the number of clock cycles. We do not need any high speed signals from external testers for testing the I/O timing parameters. The circuit architecture, circuit design details, our experimental setup and measurement results form the core of this chapter.

Chapter 4 extends the low-cost parametric measurement method explained in Chapter 3 to improve the potential limitations. With the increasing data rates, the DDR output timing specifications become more stringent and thus higher test resolution and accuracy are required. To accomplish this, we added a clock divider to generate an accurate strobe and provided a test method for the phase interpolator to verify the timing resolution. Based on these techniques, we present measurement results of the output timing parameters.

In Chapter 5, we present a delay line based timing sampler. This method is an all-digital scheme and can be easily incorporated into a current scan based delay test method to measure the time delay. In a SoC environment, this scheme is effectively used for testing high-speed parallel I/O timings with low area overhead.

Chapter 6 describes a technique to detect a small timing error for testing higher speed interface timings. We first begin with the potential shortcoming of the delay line based timing sampler, and then provide a solution of a path delay mismatch detection method. This scheme uses a charge pump based time-to-voltage converter, and a digital control method is presented to verify the time resolution. For a system-level test application, we implement an analog-to-digital converter to get digital values for test results.

Chapter 2

On-Chip Phase Interpolator Based Timing Sampler for Testing Memory Interfaces

2.1 Introduction

With the scaling of technology nodes, the operating speed of devices increased rapidly but the bandwidth of interfaces is still the major bottleneck for overall system-level performance. To circumvent the performance limitations, interface methodologies such as DDR SDRAM [39], Rambus DRAM [30], serializers/deserializers (SerDes) [72], PCI express [72], and serial ATA [72] have been developed. This dissertation focuses on issues for the design and test of the source synchronous interfaces such as DDR SDRAM and Rambus DRAM.

As the operating speed increases, the design and test issues of input/output (I/O) circuits have become critical. The key problems are the uncertain factors which cannot be exactly predicted before the tape-out. The uncertain factors influencing the I/O timing parameters are [39], a) tree-interconnect mismatch between data and clock paths [16], b) I/O pin-to-pin skew [37, 39, 68], c) impedance and trace length mismatch [48], d) power supply noise and data-dependent jitter (DDJ) [37], e) inter-symbol interference (ISI), f) duty cycle distortions, and g) a uncertain data valid window of flip-flops. Due to the uncertain factors, the valid data window becomes smaller with the increasing data rates. Thus, it has been more challenging to meet the timing specifications. The more critical issue is to test the high-speed interface timings because the I/O circuits are severely affected by the interfacing external devices. Due to the additional off-chip uncertain factors, we have been confronted with serious problems for the high-speed interface timing tests.

Traditionally, automated test equipment (ATE) is used for testing the memory I/O timing parameters such as the setup and hold time. Figure 2.1 describes the timing specifications for the setup and hold time of the memory input signals. The timing parameters are defined by the relative timing relationship between data/input and strobe/clock such as the address-setup to clock-rise (tIS), clock-rise to address-hold (tIH), data-setup to strobe-clock (tDS) and strobe-clock to data-hold (tDH). Thus, for testing the I/O timing parameters, the ATE needs to implement the proper timings among DQ, DQS, ADDR, CNRL and CLK, and perform a re-timing to align the DQS and

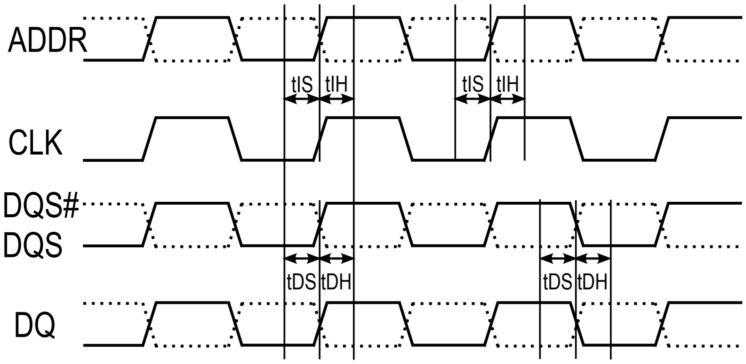


Figure 2.1: The setup and hold time for DDR memory interface

CLK [18]. The ATE also needs to generate a timing sampler to check the timing margins between data and clock [38, 60]. However, the ATE has limitations of test accuracy. Although the ATE produces data and clock patterns with high resolution, the interfacing channel devices between ATE and DUT may cause signal integrity issues due to the impedance mismatching. The off-chip parasitic components also critically affect the I/O timing parameters with the increasing data rates. Since the test equipments may not produce the same environment with the real system in which the device would be operated, the inconsistent test results of I/O timing parameters may occur. These issues exacerbate the measurement inaccuracy at wafer-level test due to more unrealistic parasitic components. Thus, we have to postpone the high-speed I/O timing test to package-level test, which hinders the decrease of time-to-market. Therefore, on-chip solutions are promising because of the higher test accuracy and smaller time-to-market.

For on-chip solutions, the I/O loopback test techniques have been presented [10, 46, 51, 61, 65]. The I/O loopback test has a structure feeding the data captured at a receiver into the input flop at the transmitter through the output and input pads. The received and transmitted data are compared and the pass/fail status is determined. This method was extended to measure most I/O DC characteristics such as the output voltages and currents (VOL, IOL, VOH, IOH), the input voltage logic levels (VIL, VIH), and the pin leakage currents (IIL, IIH, IOZ) [61]. Additionally, the characterization of jitter and noise using I/O loopback test has been proposed in [10, 46]. However, loopback tests have a fault masking problem because the interface timing of inputs and outputs is measured jointly [13]; this would degrade the accuracy of testing. In order to avoid fault masking of I/O loopback tests and to support at-speed testing, on-chip test methods have been proposed [12, 13, 15, 27, 28, 49, 55, 70]. In [28], a BIST was developed for measurement and compensation of a phase offset which is one of the timing degradation factors. An on-chip technique using a vernier delay line based time-to-digital converter (TDC) was implemented to generate the timing interval between data and clock [27, 49, 55, 70]. For memory applications, a timing generator using a delay locked loop (DLL) was implemented to test I/O setup and hold time [12, 13].

One of key challenges for testing high-speed parallel I/O timings generates the appropriate timing delay between data and clock [31, 56]. Therefore, this chapter presents an on-chip timing sampler to achieve this goal and avoid the limitations due to off-chip test methods [36]. The on-chip timing sampler combines an on-chip pattern generator and a sampling clock (strobe) generator. This scheme generates and controls a timing delay between data and clock, and is also used to measure the timing difference. In order to compare our work with off-chip test methodologies, we first review off-chip test circuits for testing memory interface timings in the next section.

2.2 Survey of Off-Chip Timing Test Methods

In this section, we review previous methods for testing source synchronous memory interface timings to explain the fundamental test methodologies. Since it takes longer time to newly develop an advanced ATE, previous test methods focus on ATE-aided solutions. The built-off circuits add new functions to the existing ATE to improve accuracy and speed performances. For testing the source synchronous memory interface timings, the ATE first needs to build a certain timing relationship between the data coming from DUT and the tester clock. The test flow thus starts to search the DQ/DQS signal edges with respect to the clock period of ATE. Therefore, the typical test flows are as follows [60, 71]:

- Set the strobe-clock timing for sampling DQ and DQS
- Run a test pattern
- Determine PASS or FAIL
- Shift the strobe-clock timing
- Repeat the above three processes until the strobe scanning is completed
- Detect the transition timings for DQ and DQS
- Calculate the time delay difference between DQ and DQS

Based on these test flows, an off-chip circuit for testing data skews for memory devices has been developed as shown in Figure 2.2 The ATE separately generates the strobes to capture DQ and DQS coming from the DUT. The strobes are shifted to detect the transition timings of DQ and DQS while the strobes are scanning. Once the transition timings of DQ and DQS are detected, data

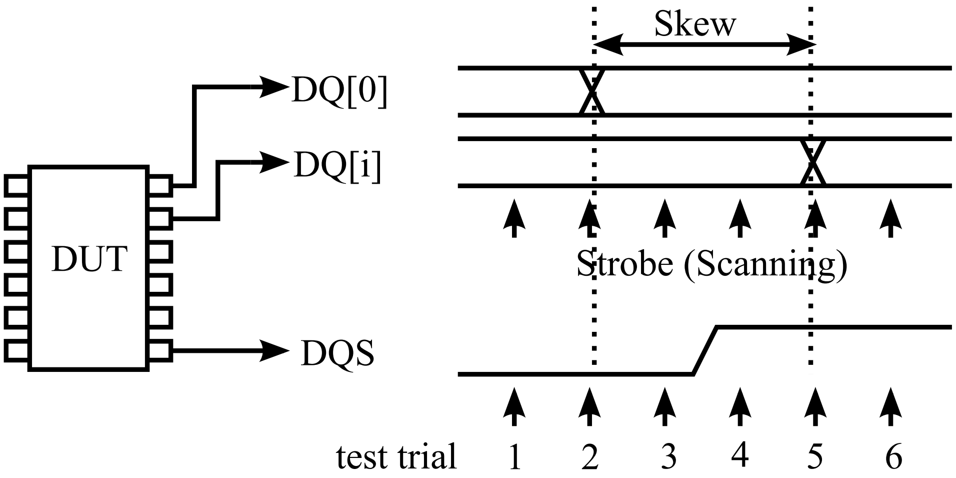


Figure 2.2: Conventional data skew test method

skews are measured by calculating the time delay among the transition timings. However, there are two issues for this conventional source-synchronous test method. This method takes a long test time because the ATE runs a test pattern repeatedly while the strobes are scanning. Thus, the test time increases proportionally to the number of the strobe-scanning. For testing higher speed timings requiring high resolution, the test time becomes much longer. Moreover, each timing jitter of DQ and DQS is summed because this method does not detect the transition timings of DQ and DQS in the same test cycle. The timing jitter significantly causes a measurement error for testing high data rates reaching over 1 Gbps.

In order to improve test time and reduce the measurement error, a multi-strobe circuit has been developed as shown in Figure 2.3 [71]. This technique generates multiple strobe phases and detects the transition timings of DQ and DQS during a single test cycle. Therefore, the common-

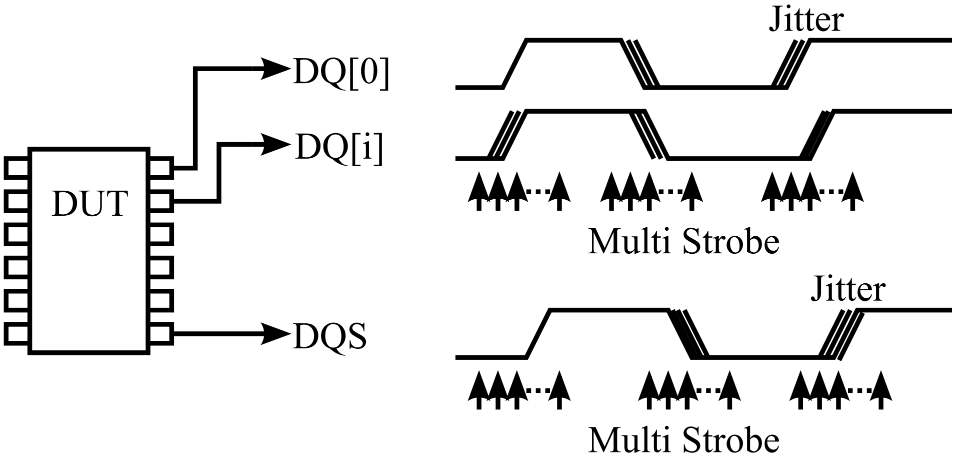


Figure 2.3: Data skew test method using multi strobe circuit

mode jitter between DQ and DQS is canceled, and the test time decreases by '$\frac{1}{\#\text{ of the strobe-edges}}$' because the ATE does not need to run the same test pattern repeatedly.

Another critical issue of ATE is to implement the strobes with high accuracy and resolution. Figure 2.4 shows the new multi strobe circuit using a delay-locked loop (DLL) based delay line [71]. The DLL circuit produces a sampling clock with a ±20 ps resolution and ATE can thus test memory output timings.

Another critical component for testing high data rates is a timing generator. To obtain a high test quality, at-speed testing is required, but it is limited by a tester clock speed. Thus, it becomes necessary to implement a high-speed and high-precision timing generator. A conventional timing generator consists of RAM and a variable delay block as shown in Figure 2.5(a). Figure 2.5(b) shows an implementation method for a 1.066 GHz timing generator using the

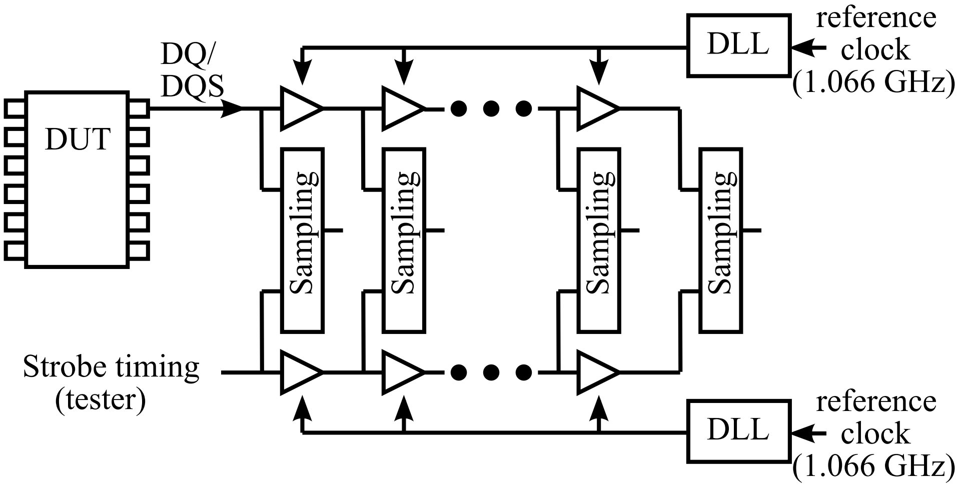
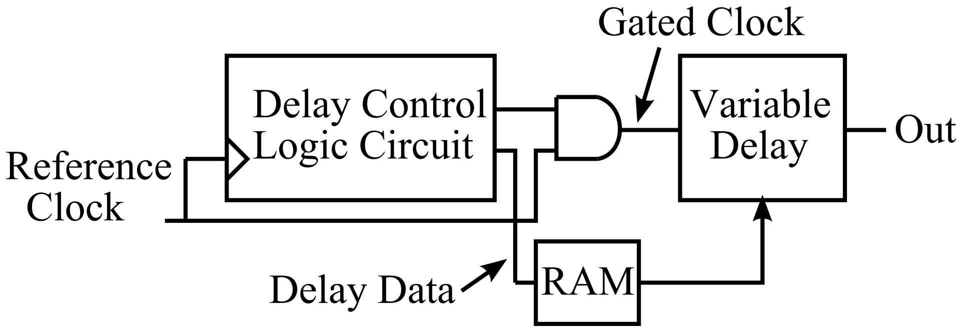


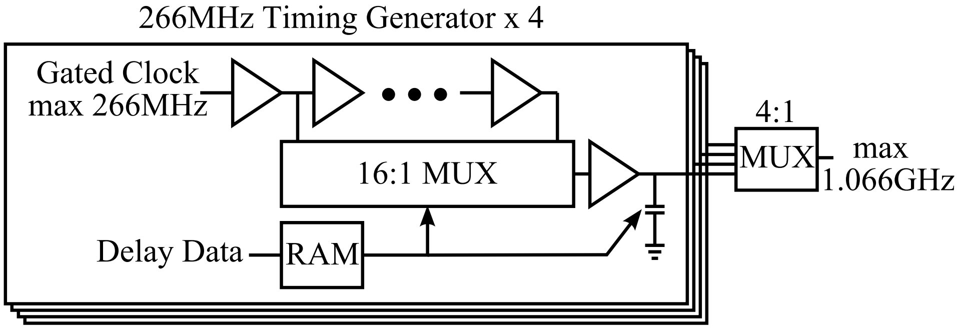
Figure 2.4: DLL based multi strobe circuit

previous conventional timing generator (Figure 2.5(a)). The 1.066 GHz timing generator is implemented using four 266 MHz timing generators and one 4:1 timing multiplexer. However, the performance of the conventional technique has constrained by the RAM speed. Moreover, the requirements of a higher precision need bigger RAM sizes, which increases test costs. Thus, a timing generator without RAM becomes necessary. As shown in Figure 2.6, the timing generator using a digital DLL and a fine delay circuit has been developed [42]. Using this architecture, the 1.066 GHz timing generator was implemented as shown in Figure 2.6 and also the 4.266 GHz timing generator was designed using four 1.066 GHz timing generators, two 2:1 MUX and S-R latch.

The timing generator and the multi strobe circuit we introduced are the key components for testing high-speed memory interfaces. Figure 2.7 presents



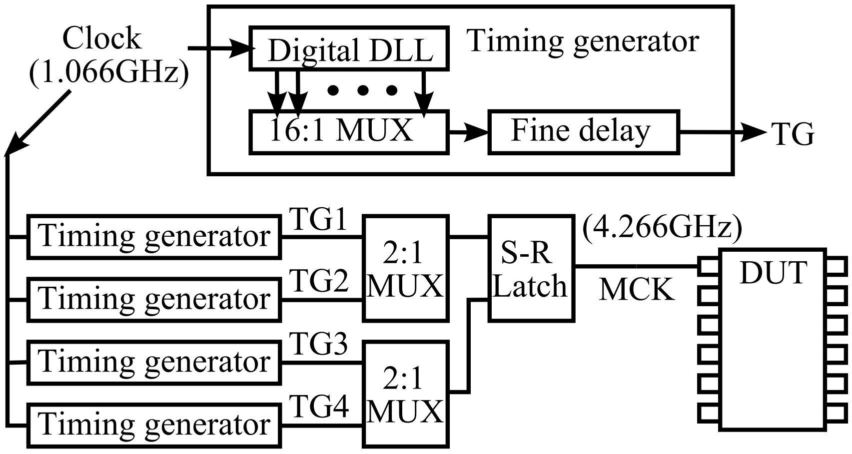
(a) The block diagram of a conventional timing generator



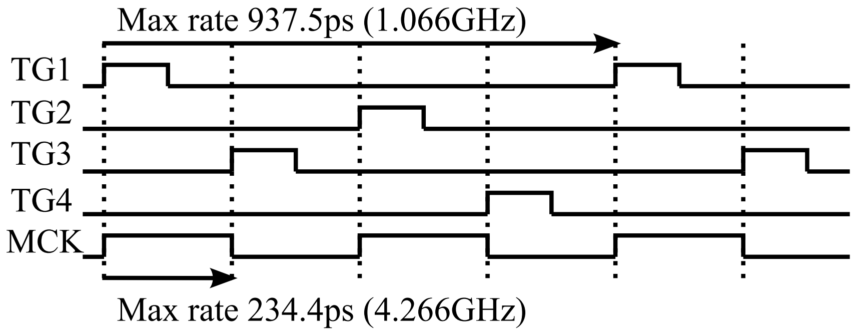
(b) Implementation of 1.066GHz timing generator

Figure 2.5: Conventional timing generator

a memory test system by combining the main circuits shown in Figure 2.4 and Figure 2.6. The timing generator provides a triggering signal with a dynamic arbitrary frequency and phase. A vernier delay line based time-to-digital converter (TDC) detects the transition timings of DQ and DQS, and quantizes the interval between the DQ/DQS and the triggering signal. Using two TDCs, each time interval of DQ and DQS with respect to the triggering signal is determined. Thus, the test scheme measures the time delay difference between DQ and DQS coming from memory devices. Based on the time delay, this tester can measure timing parameters for memory



(a) 4.266 GHz timing generator



(b) 4.266GHz timing diagram

Figure 2.6: Architecture for 4.266GHz memory test system

devices. This test circuit was implemented to test 2 Gbps memory devices with a variable resolution of 10 ps to 40 ps [70].

However, off-chip test methods have limitations of test cost and signal integrity. Our goal is to implement on-chip test methods to take benefits of the off-chip test methods and avoid the constraints. From the existing off-chip test methods, we understand that the crucial components are an on-chip

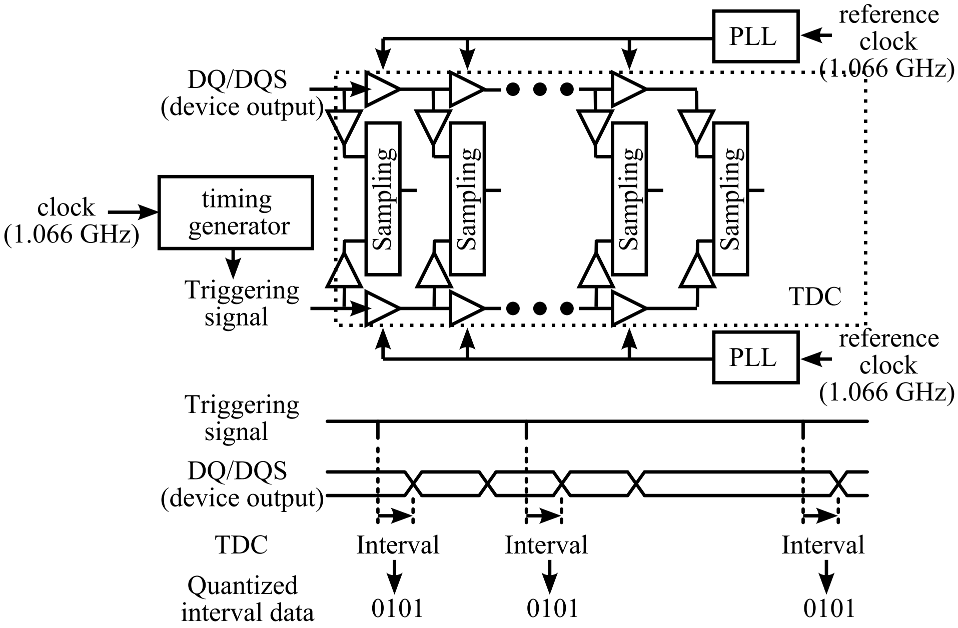


Figure 2.7: TDC based test circuit for memory test systems

signal generator and a strobe generator and the key challenges are to obtain a smaller measurement error and a higher resolution. To accomplish this, we explore our on-chip timing sampler including both on-chip pattern generator and strobe generator in the next section.

2.3 Circuit Implementation

Figure 2.8 shows the overall block diagram of the on-chip timing sampler. The main components of this scheme are 1) four phase generator, 2) coarse control unit for phase shifting, 3) fine control unit for precise controls of the generated phases, 4) pulse control unit for the sampler generation.

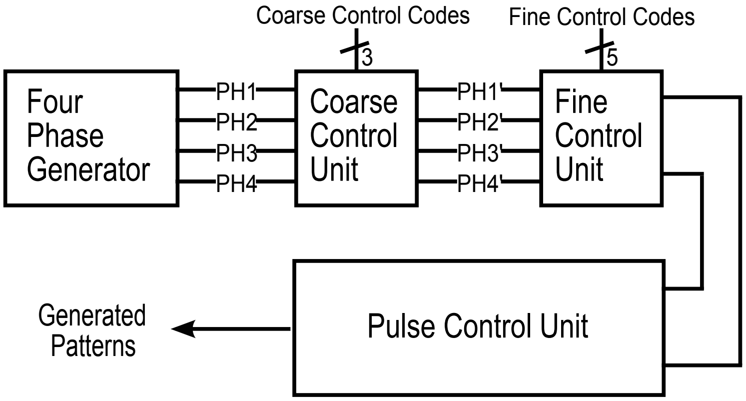
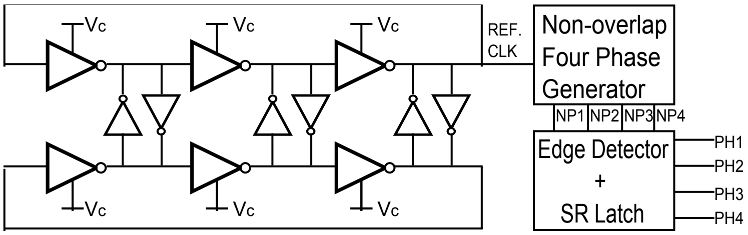


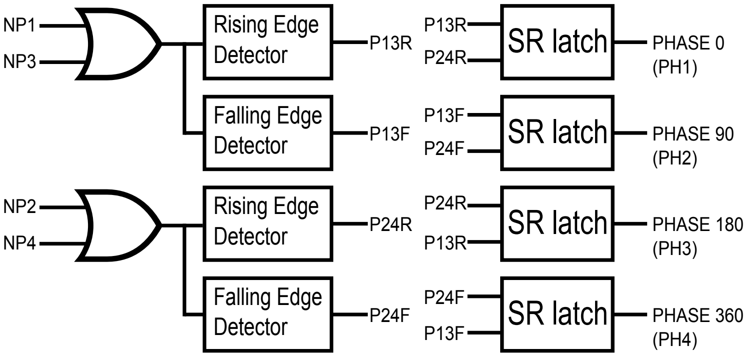
Figure 2.8: Overall structure of on-chip timing sampler

2.3.1 Ring Oscillator Based Four Phase Generator

The four phase generator is shown in Figure 2.9. The four phase signals of 0°, 90°, 180° and 270° need to be generated to control the edges of clock and data patterns. This function is implemented with a ring oscillator based VCO, falling and rising edge detectors, S-R Latches and a non-overlapping four phase generator. By using this circuitry for both data and clock signals, the errors due to an external ATE are avoided, and mismatches between clock and data paths, generated only on-chip, can be detected. The VCO generates a periodic signal whose frequency is controlled by the tunable control voltage level [59]. The gain of the VCO is determined depending on the required operating frequency range in a given system. This circuitry is used for both data and clock patterns so that the two signals have the same noise source. Figure 2.9(a) shows the ring-oscillator based VCO. Its output, REF CLK, is fed into the non-overlapping phase generator designed using flip-flops and decoders to generate four-phase signals – NP1, NP2, NP3 and NP4. Figure 2.9(b) shows



(a) Overall structure



(b) sub-circuits (edge detector and SR latch)

Figure 2.9: Four phase generator

the sub-circuits for the four phase generator. The edge detectors detect the rising and falling edges of NP1, NP2, NP3 and NP4 and the SR latches form the four phase signals – PH1, PH2, PH3 and PH4. Thus, this scheme generates four phases and the overall circuit operations are shown in Figure 2.10. The edge locations of the generated phases can be changed by the following control units to generate various data patterns.

2.3.2 Coarse Control Unit for Phase Shifting

The generated four phases are shifted by the coarse control unit. Figure 2.11(a) shows the overall structure for the coarse control unit. The various

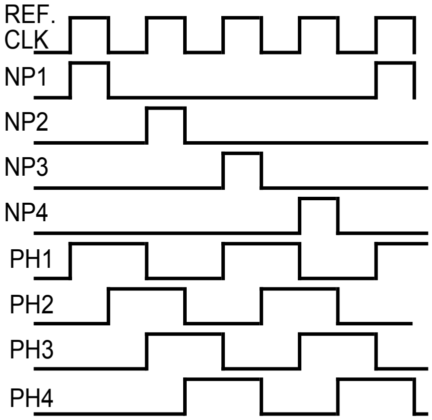
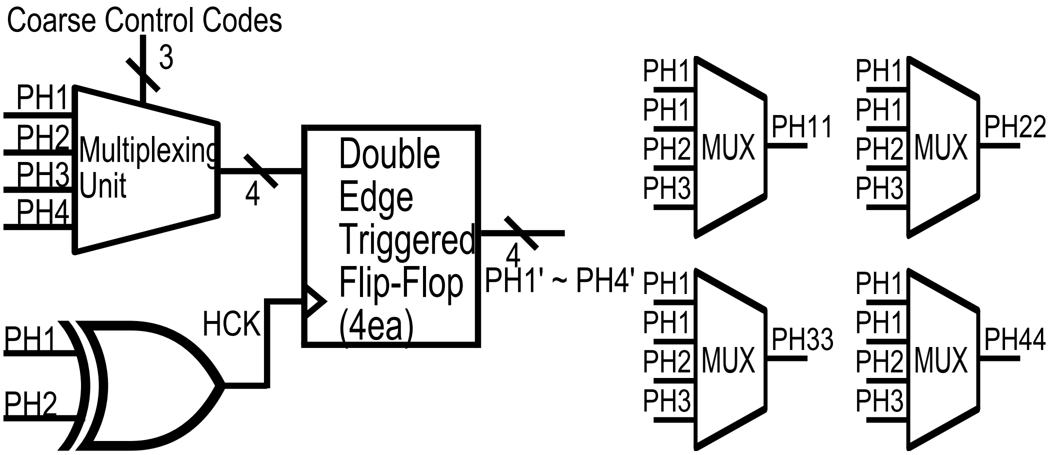


Figure 2.10: Circuit operation for four phase generator

data patterns can be generated by controlling the rising and falling edge positions. Thus, the objective of the coarse control unit is to determine coarsely the rising and falling edge positions of four signals – PH1′, PH2′, PH3′ and PH4′. The PH1′ and PH2′ are used to generate the rising edges of clock and data, and PH3′ and PH4′ are used to generate the falling edges. The PH1′ and PH2′ are used to generate the rising edges and thus the rising edges of PH1′ and PH2′ should come before the PH3′ and PH4′. The multiplexing unit shown in Figure 2.11(b) are designed such that the sequence of the phase shifts has a definite order. Table 3.2 shows the definite order for the phase shifts according to the coarse control codes. For example, if the coarse control code is ‘000’, the edges of PH1′, PH2′, PH3′ and PH4′ are changed to PH1, PH2, PH2 and PH3, respectively. Thus, the rising and falling edges of the generated pattern are located between PH1 and PH2, and PH2 and PH3, respectively. The double-edge triggered flip-flops, one of sub-circuits for the coarse control unit, are used to remove glitches occurring in the outputs of the multiplexing



(a) Overall structure (b) Sub-circuit (multiplexing unit)

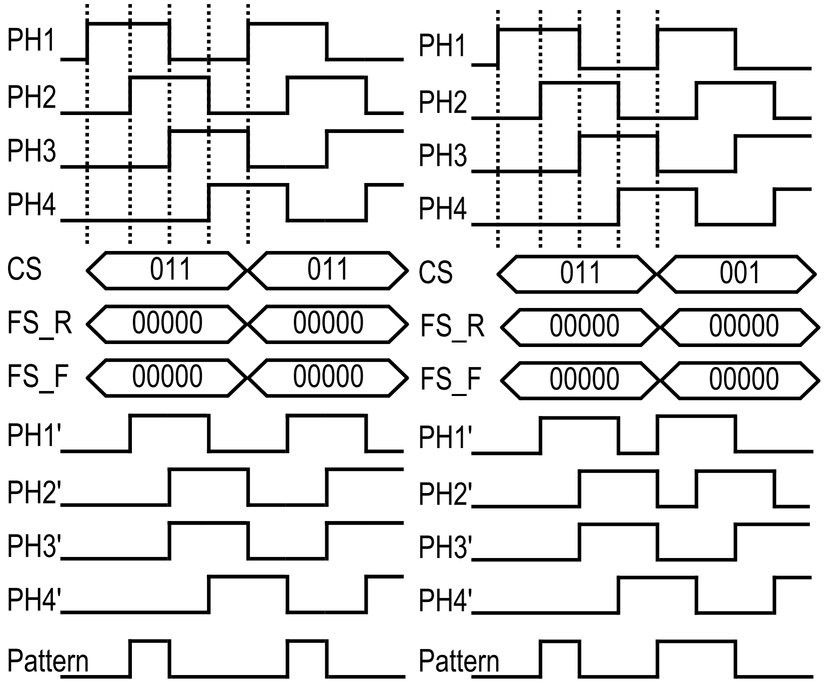
Figure 2.11: Coarse control unit

control codes			after phase shifting				edge location	
CS2	CS1	CS0	PH1′	PH2′	PH3′	PH4′	rising edge	falling edge
0	0	0	PH1	PH2	PH2	PH3	PH1 to PH2	PH2 to PH3
0	0	1	PH1	PH2	PH3	PH4	PH1 to PH2	PH3 to PH4
0	1	0	PH1	PH2	PH4	PH1	PH1 to PH2	PH4 to PH1
0	1	1	PH2	PH3	PH3	PH4	PH2 to PH3	PH3 to PH4
1	0	0	PH2	PH3	PH4	PH1	PH2 to PH3	PH4 to PH1
1	0	1	PH3	PH4	PH4	PH1	PH3 to PH4	PH4 to PH1

Table 2.1: Table showing coarse control codes for phase shifting

unit.

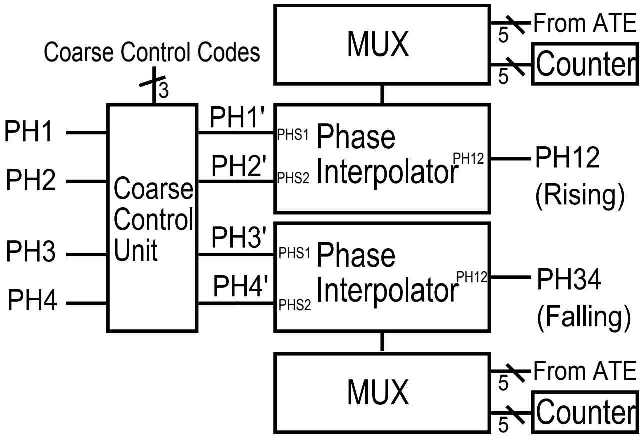
Figure 2.12(a) and 2.12(b) show two examples of the coarse control timing. The coarse control code (CS) is asserted during one phase cycle and can be changed at every phase cycle. Figure 2.12(a) describes the case of ‘CS=011’, which shows the edges of the PH1′, PH2′, PH3′ and PH4′ are changed to PH2, PH3, PH3 and PH4, respectively. Figure 2.12(b) describes the case that CS is ‘011’ in the first phase cycle and then CS is changed to ’001’ in the second



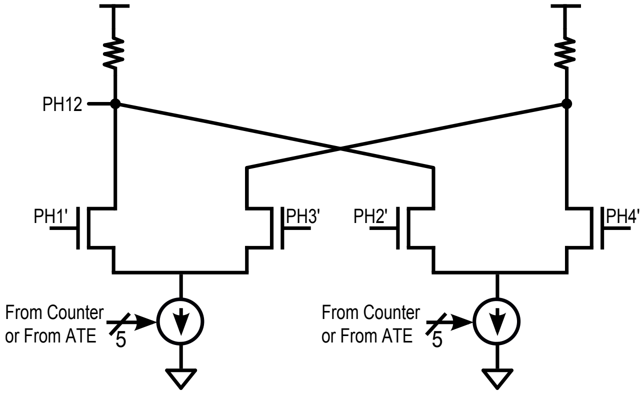
(a) CS='011' (b) CS='011'/'001'

Figure 2.12: Coarse control timing

phase cycle. Thus, the generated phases are the same as the case shown in Figure 2.12(a) during the first cycle. On the other hand, during the second phase cycle, the edges of the PH1′, PH2′, PH3′ and PH4′ are changed to PH1, PH2, PH3 and PH4, respectively. Hence, this scheme can control the pulse width and the edge locations of clock and data patterns at every clock cycle, and we name it a cycle-by-cycle control method.



(a) Overall structure



(b) Phase interpolator (PI)

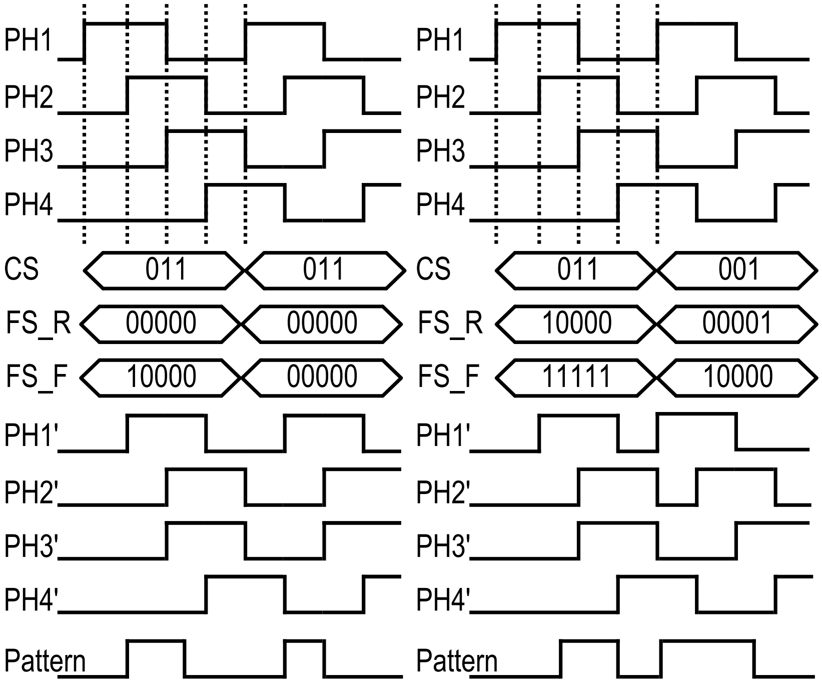
Figure 2.13: Fine control unit

2.3.3 Fine Control Unit and Pulse Control Unit

Figure 2.13 shows the overall structure for the fine control unit. The objective of this block is to precisely control the rising and falling edges of the generated phases. The fine control unit consists of the multiplexor and the phase interpolator (PI). The PI shown in Figure 2.13(b) is the main circuit for the fine control. The resolution of the PI is controlled by the current sources and the currents are controlled by the fine control codes. Depending on test methods, we obtain the fine control codes from testers or internal counters. If we use the internal counters, we automatically generate the codes and self-test is available. In this chapter, we use 5-bit control codes for the fine control code, and thus we obtain the resolution of 'tCK/128'. Since we do not need to test with high resolution for low-speed memory devices, the tCK based variable resolution is acceptable. Moreover, we separately control the rising and falling edges of the generated patterns by using two phase interpolators. Each PI is independently used to control the rising or falling edge, and thus one PI uses the PH1$'$ and PH2$'$ as inputs to control the rising edges and another PI uses PH3$'$ and PH4$'$ as inputs to control the falling edges. The interpolated signal, PH12, determines the rising or falling edge placement.

Figure 2.14(a) and Figure 2.14(b) show two examples of the fine control timing where FS_R and FS_F indicate the fine control codes to control the rising and falling edges, respectively. The two fine control codes can be separately assigned. These two examples show that our scheme generates more various patterns, and control the phase edges precisely and thus the generated pattern can also be used as a sampler. Moreover, when we use the internal counters to generate the fine control codes, this scheme controls the phase edges at every clock cycle - cycle-by-cycle control method.

Figure 2.15 shows the pulse control unit which consists of the edge detectors and SR latches. This unit simply forms the pulse for the pattern using the interpolated signals, PH12. The two 'PH12' indicate the rising and



(a) FS='10000'/'00000' (b) FS='11111'/'10000'

Figure 2.14: Fine control timing

falling edges, respectively, and the pulse control unit forms the pattern to obtain data or strobe signal.

2.4 BIST Application

A BIST method is preferred to avoid high-speed signalling between ATE and DUT. The on-chip timing sampler is thus implemented as BIST applications. The BIST scheme can be incorporated in a memory device for testing the interface timing parameters. This section describes an example of

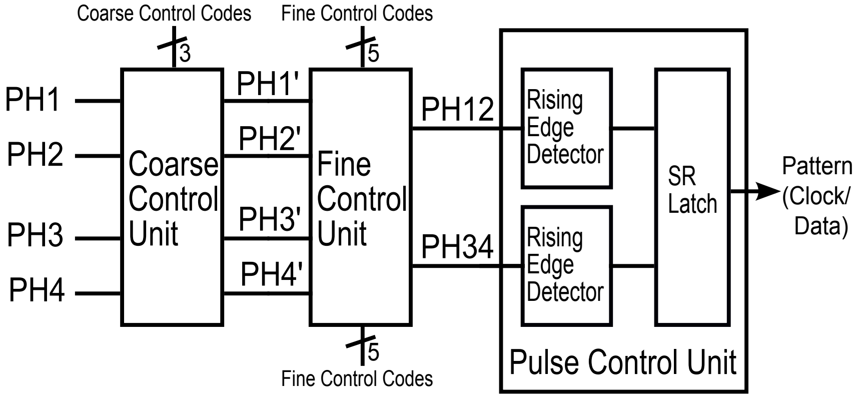


Figure 2.15: Pulse control unit

application and presents the BIST structure for the application.

2.4.1 An Example of Application

The DQ and DQS are bidirectional signals and are outputs and inputs for OUTPUT REG and INPUT DATA REG, respectively, and thus the DQ and DQS are used for both the read and write operations. Using the BIST scheme including the on-chip timing sampler, we can insert delay mismatches between data and strobe for testing input timings during the memory write operation. We also measure the delay mismatches for testing output timings during the memory read operation. Thus, using this BIST scheme, we test the input setup and hold times, voltage input low (V_{IL}) level, voltage input high (V_{IH}) level, duty cycle tolerance, and data skews.

In order to measure the I/O timing parameters, the BIST scheme can be incorporated in a DDR memory architecture to test memory I/O timing parameters as shown in Figure 2.16. The tri-state buffers are used to switch

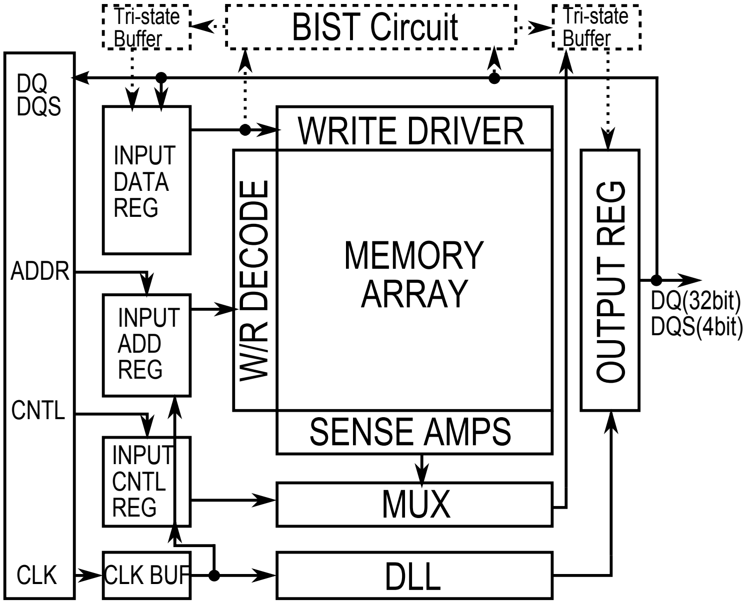


Figure 2.16: An example of memory architecture

from normal to test modes. In the normal mode, the INPUT DATA REG gets data from external testers or adjacent systems. In the test mode, the tri-state buffers disconnect the normal signal paths from I/O registers and connect the BIST signals to the I/O registers. The data and clock paths of I/O registers are designed to have the same delay by implementing perfect data and clock trees. However, mismatches occur due to uncertain factors such as jitter, process variations, etc [37]. Thus, we detect such mismatch factors and measure the delay mismatches since the mismatches critically affect the I/O timing parameters. Using the on-chip timing sampler, we generate the data and strobe, and also measure the delay mismatch between data and strobe.

2.4.2 Circuit Architecture

For the source synchronous memory devices, the DQ and DQS are propagating in parallel and thus the relative timing margin is critical and directly affect the I/O timing specifications. Thus, the BIST scheme also should be designed to be matching structures between data and clock patterns. The BIST scheme is thus configured for testing memory I/O timing parameters as shown in Figure 2.17. The BIST circuit is implemented and incorporated into the normal circuit on-chip. This scheme requires input DC signals from

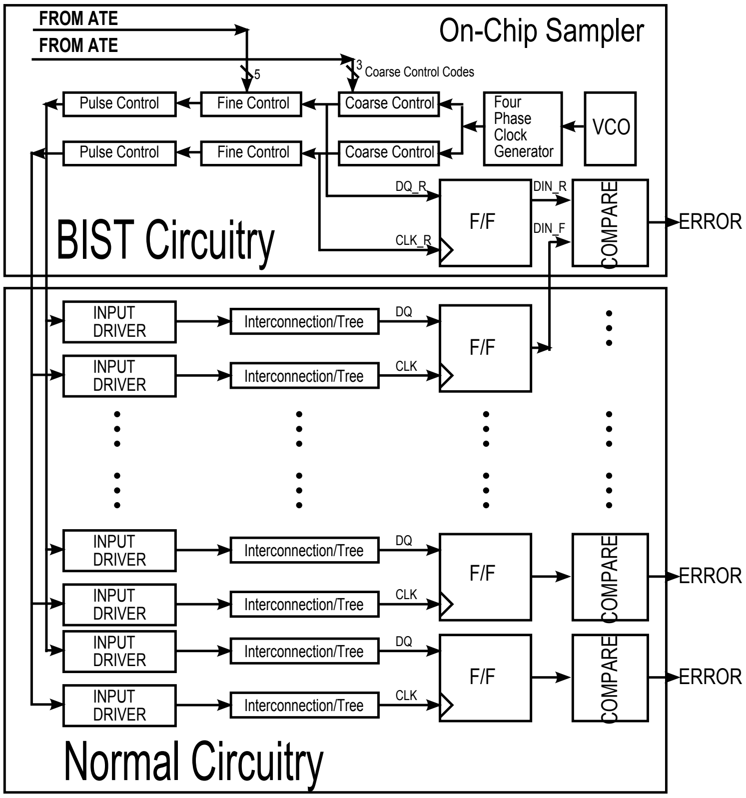


Figure 2.17: An example structure for BIST

testers to control internal circuitry and then determine pass or fail by monitoring the BIST outputs. The self-testing is initiated by the coarse and fine control codes and is completed by comparing results. In this BIST circuit, the generated clock and data signals are fed into input drivers in the normal circuit to analyze the degrading effects on I/O performance due to the variations of input drivers and mismatch tree-interconnect. The path generating 'DQ_R' and 'CLK_R' in the BIST circuitry is called the 'REFERENCE PATH'. The path generating 'DQ' and 'CLK' in the normal circuitry is called the 'MEASUREMENT PATH'. The outputs of two paths are compared to determine pass or fail for the I/O timings. The input timings of the two flip-flops in the reference and measurement paths are separately controlled using different coarse and fine code values. The input timings of the flip-flop in the reference path is controlled to have sufficient margins. On the other hand, in the measurement path, the input timings of the flip-flop are set by the specifications of the setup and hold time. If the flip-flop is failed, we determine that the input timings for the memory device do not meet the timing specifications.

2.4.3 Overall Operation

This section describes the overall operation of the BIST structure which includes the four phase generator, coarse and fine control, and comparison - on-chip timing sampler. Figure 2.18 shows the basic operation of the BIST circuit.

As shown in Figure 2.17, the output results of two flip-flops are com-

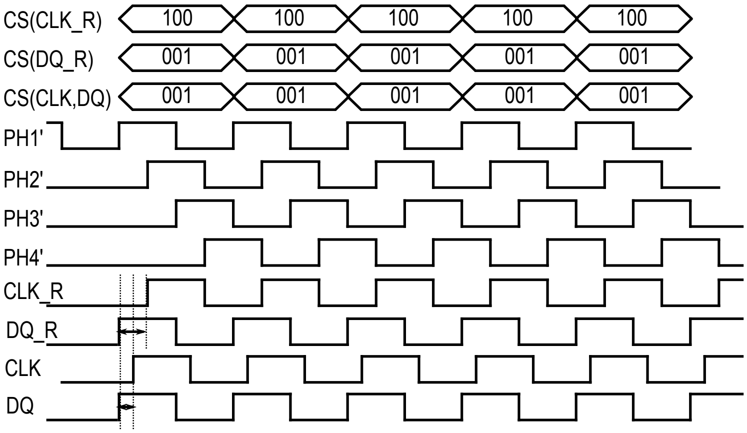


Figure 2.18: Overall timing operation

pared to decide on pass or fail. The flip-flop in the reference path has enough margin between ‘DQ_R’ and ‘CLK_R’ such that there is no failure in capturing data. Since detailed tuning is not required for this path, only units for coarse control are necessary to control circuits in the reference path. On the other hand, for the measurement path, the delay difference between ‘DQ’ and ‘CLK’ is given by the specification of setup and hold times. Circuits in the measurement path need to be controlled by using both coarse and fine control units to represent the required specification. For the basic operation of the BIST circuit, the CS and FS codes are not changed at each cycle and are given from an external tester. Finally, the results of two paths are compared to find if the system meets the required specification. In Figure 2.18, it seems that the two flip-flops have enough margin of the setup and hold times, but the outputs of flip-flops can have wrong values due to uncertainties of flip-flops and internal mismatches between clock and data paths.

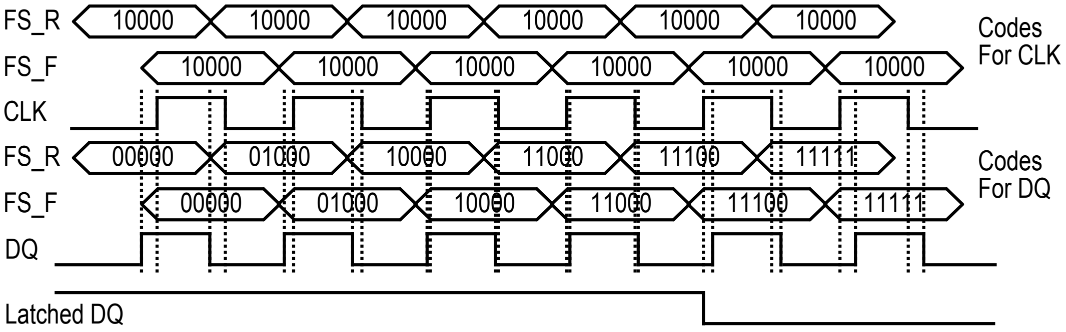


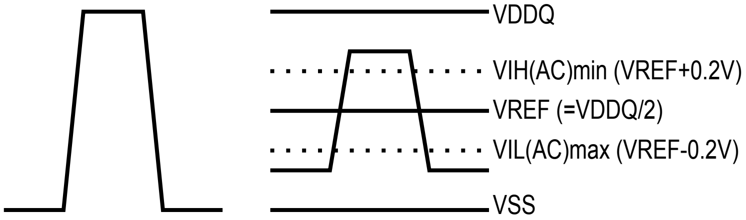
Figure 2.19: Extended timing operation

The BIST scheme can be applied to analyze the effect of mismatches on I/O performance by using the extended control method. In the extended operation as shown in Figure 2.19, the control method used in the reference path is still the same as the basic operation shown in Figure 2.18. On the other hand, control codes for circuits used in the measurement path need to be configured. Instead of using fixed FS codes from ATE, the embedded counter is used to control the location of edges cycle-by-cycle. The initial value of the counter is set '00000' or '10000' depending on the required patterns, and then the output of the counter moves to the direction of 'UP' or 'DOWN'. As shown in Figure 2.19, there is a failing point when latched data is changed. The failing points can be different for different I/O pins. Therefore, we can detect if per-pin skew creates severe effects on I/O performance by applying the BIST scheme to all I/O paths. For example, in case of a DDR interface with 32 data (DQ) pins, if the latched data is '11111000' for DQ1 and '10000000' for DQ30, the latched values are compared with the generated value from the reference path. As shown in Figure 2.17, this circuit will generate the comparison results and ‘ERROR’ signals, corresponding to the amount of internal mismatch. We monitor the number of cycles of ‘ERROR’ signals to measure how much this system violates the I/O specifications of the setup and hold times. The ‘ERROR’ signal has a unit of ‘one cycle time * the resolution of the phase interpolator’. Since ATE may not have the resolution to monitor signals with high frequency and accuracy, we generated ‘ERROR’ as a clock based signal. Additionally, we can generate different ‘DQ’ patterns at each clock cycle as shown in Figure 2.14(a) and 2.14(b). By applying various patterns to the BIST circuit, the degradation effects caused by data-dependent jitter [41] can be analyzed and the worst-case data pattern can be detected.

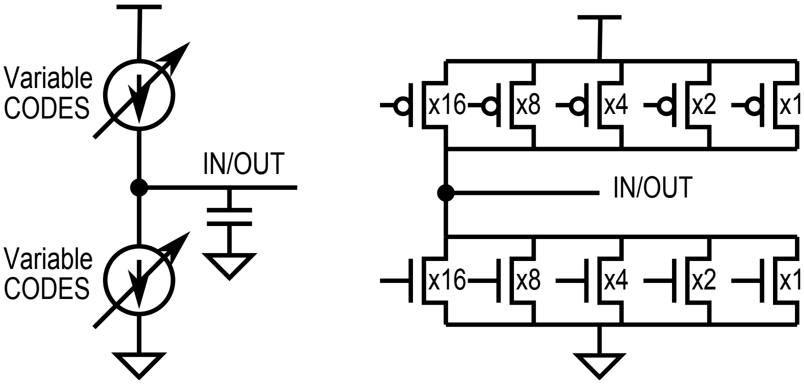
2.4.4 Setup and Hold Times

Using the BIST scheme, we test input setup and hold time by measuring the relative timing difference between DQ and DQS. For the ‘REFERENCE PATH’, we set the coarse and fine control codes to the required timing specifications. For the ‘MEASUREMENT PATH’, the phase edges are precisely controlled using the cycle-by-cycle control method. The outputs of two paths are then compared. We do the same test for all other DQ pins and then measure the setup and hold times considering the performance degradations due to input data skews. Moreover, this work extended the BIST scheme for testing the timing degradations due to input voltage-level variations and duty cycle distortions.

2.4.5 AC Input Voltage-level Variation Tolerance



(a) AC input-level definition



(b) Circuit operation (c) Circuit implementation

Figure 2.20: Voltage-level control scheme

One of I/O specifications is the input voltage-levels in DDR memory devices [1]. Figure 2.20(a) shows the definitions of V_{IL} and V_{IH}. As can be seen, the AC input voltage-level is specified in the range of 'V_{REF}-0.20V' to 'V_{REF}+0.20V', where the V_{REF} is the reference voltage of input receivers in single-ended memory devices. The voltage level is one half of the power supply voltage in DDR2 memory. Because of the impedance mismatching of board and channel, the input-levels have variations. Therefore, it is hard to test the I/O parameters. We present the function to test I/O parameters even under the variations. While the input-level is precisely changed, we read out the output of SAFF to test the amount of effects on I/O performance. Figure 2.20 shows the operations and the circuit diagrams of the voltage-level control scheme. The voltage-level control scheme is implemented with a current steering logic (CSL) as shown in Figure 2.20(b). The CSL is designed with the segmented current-steering type using 5-bit control codes as shown in Figure 2.20(c). The amount of currents is precisely controlled. This circuitry can be controlled in two ways - a symmetric-mode and an asymmetric-mode. In the symmetric-mode, the strengths of the PMOS driver and the NMOS driver presented in Figure 2.20(c) are controlled equally by using the same code values. In the asymmetric-mode, the different code values can be applied to control independently the driver strengths. In the real operations, since the V_{IL} and the V_{IH} can be varied asymmetrically, we need to inspect the effects on I/O parameters due to the mismatches. Additionally, the fix-mode and self-mode are also set using MRS codes and internal counters. Using the procedure, we explored the tolerances of the internal I/O circuits to voltage-level variations. We read out the code values at the transition points of the 'DQ' signal. The maximum code difference between the transition points is corresponding to the delay mismatch which indicates the timing variations due to input voltage-level distortion.

2.4.6 Duty-cycle Distortion Tolerance

Since the data path in the high speed memory operate in DDR mode, the duty-cycle distortions significantly affect I/O timings. The distortions are caused by inaccurate external signals. Besides, the internal I/O circuitry including input receivers, input sense-amplifiers, and path tree interconnections also produce the duty-ratio distortions. Using the BIST circuit, the effects due to the distortions on I/O parameters are investigated. The BIST scheme controls independently both rising and falling edges using the cycle-by-cycle control method. We set the ideal condition of 50% duty-ratio to the measurement path. On the other hand, we give duty-ratio distortions to the reference path by controlling differently rising and falling edges. Under the duty-ratio distortions, we measure the relative timing variations between two paths by monitoring 'DQ' signals.

2.5 Simulation Results

We have designed the BIST circuitry using a 0.18-μm CMOS process and simulated at the transistor-level using Hspice. All simulations are performed at the typical process corner, room temperature, and supply voltage of 1.8V. Monte-Carlo simulations are performed using the statistical process parameters given by the 0.18-μm TSMC CMOS process.

Figure 2.21 shows simulation results for the frequency range of the reference clock using a ring oscillator based VCO. We achieved a frequency range from 500MHz to 2GHz for the voltage range from 0.83V to 1.8V. Since clock and data signals are generated with half the frequency of the reference clock, the BIST circuit can be operated from 250MHz to 1GHz. However, we can tune the ranges by controlling the voltage of the ring oscillator.

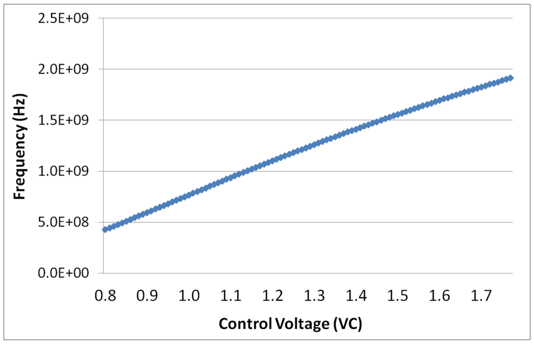
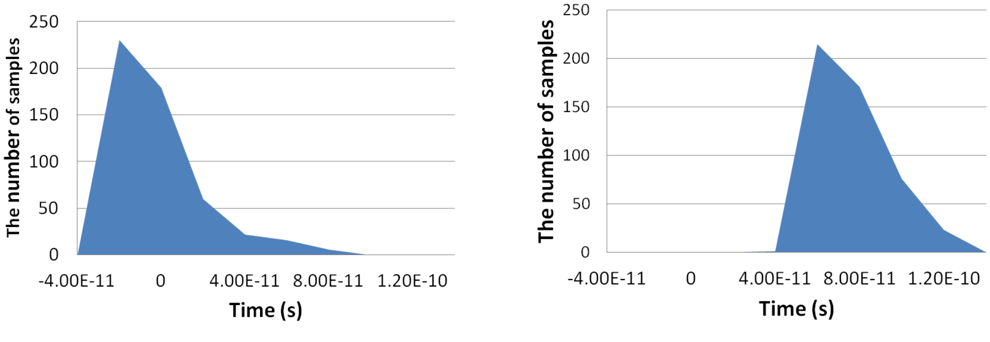


Figure 2.21: Control voltage vs. Frequency range

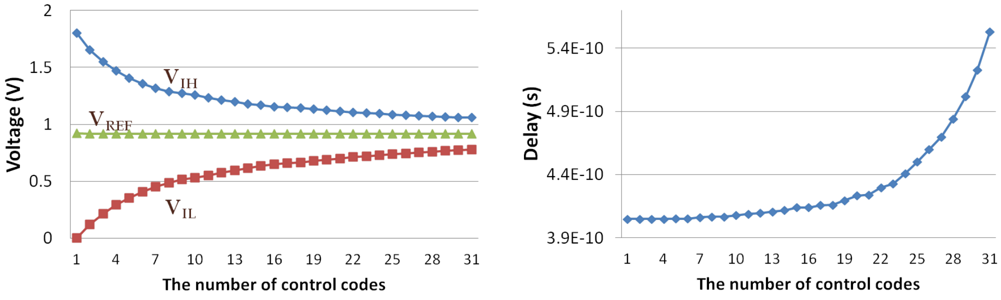
Figure 2.22 shows the Monte-Carlo simulation results for the setup and hold time. The edges of clock in the reference path are calibrated. We measure the delays between data and clock at the rising and the falling edges to specify the setup and the hold times when the results have valid values. Figure 2.22(a) and 2.22(b) show the histogram for the simulation results. As a result, the setup and hold times have the mean values of -2ps and 50ps, respectively. However, the results will be worse if we include the effects caused by data patterns, voltage-level variations, and duty-cycle distortions.



(a) Setup time

(b) Hold time

Figure 2.22: Monte Carlo simulation results for setup and hold time



(a) V_{IH}, V_{REF}, and V_{IL} (b) Input-level variations vs. delay

Figure 2.23: Voltage-level tolerance simulation

Figure 2.23 shows the simulation results for testing voltage-level tolerance. The value of the V_{REF} is 0.9V, which is one half of the supply voltage of 1.8V, and the V_{IL} and the V_{IH} are varied in the range of '$V_{REF}\pm0.10$' to '$V_{REF}\pm0.9$'. While we change the input voltage-levels, we test if the I/O parameters are still passed. For the simulations, we fixed the input voltage-level to '$V_{REF}\pm0.20$' in the measurement path. On the other hand, we calibrate the input voltage-levels of the reference path. Such variations cause the relative timing differences between data and clock. If the timing interval is too large, the timing fails are occurred. In order to give the input-level variations, we use the 5-bit control codes. According to the codes, the simulation results of the voltage-level variations are shown in Figure 2.23(a). While the input-level is changed, the output of the SAFF is monitored to test the effects due to input-level variations. As shown in Figure 2.23(b), the propagation delays drastically increase at the marginal point, which causes the timing fails.

2.6 Conclusion

For testing the setup and hold time of source-synchronous memory I/O parameters, an on-chip timing sampler has been developed. This technique generates data and clock, and precisely controls the time interval between data and clock at every tCK cycle. The on-chip timing sampler is implemented for the BIST application and the BIST structure is incorporated in a memory architecture for testing memory I/O timing parameters. The embedded pattern generator avoids the timing degradations due to the input level distortions. Moreover, we insert timing jitter by giving input-level or duty distortions to check if the memory device has enough timing margins or tolerances. The timing controller enables to measure the timing margin difference between reference and measurement paths. Using this scheme, we check if the memory device meets the timing specifications for the setup and hold times by comparing the results between two paths. The generated timing edges are controlled at every clock cycle, and thus this method considers power fluctuations and is able to estimate the worst timing margin.

Chapter 3

Low-Cost Measurement Methodology for Testing Source Synchronous Interface Timing

3.1 Introduction

A source-synchronous interface technique is a data-transferring method to achieve high bandwidth, where data and the dedicated strobe-clock are transferred in parallel from a transmitter to a receiver, and the receiver interface uses a clock to latch the transferred data. The source synchronous structure improves bandwidth by avoiding the limitations due to the time of flight through interconnects between chips. On the other hand, since a traditional ATE has no functions for source synchronous interface timing tests, testing issues become more critical with the increasing data rates. Thus, the ATE needs to implement the strobe signal to capture the data transferring from the DUT. The implementations of ATE for the source synchronous functions increase hardware complexities. Since high-speed I/O timing tests are constrained by the speed and accuracy of tester clocks, the requirements of the high edge placement accuracy (EPA) and high resolution drastically increase test costs [11]. As the data rates approach multi-gigabps, the test issues become more critical because the data window is narrower but the timing jitter and PVT variations are not decreasing. Moreover, the timing specifications such as the setup and hold times and output skews for the DDR memory get tighter by around 100ps [2] for over 1GHz. Thus, it has become more challenging to find the valid data window because the timing jitter takes a large portion of the data bit width. To detect the narrow valid data window, ATE need to implement high resolution and high edge placement accuracy. The more critical issue is that ATE has not been developed at a faster rate than the device technology [44]. The low-cost ATE has limitations for controlling input voltage-levels, edge placements and duty ratios with high accuracy and precision. These factors make I/O timing measurements very challenging.

This chapter thus explores a low-cost measurement method for testing source synchronous interface timings by supporting testability with the existing low-end ATE [32]. In order to validate the low-cost test scheme, we implemented a prototype chip including the on-chip timing sampler presented in Chapter 2. This test chip does not need any high-speed interface signals from testers using the on-chip timing sampler and thus a conventional low-cost test equipment is enough to test this chip. Based on the low-cost measurement method, this chapter presents a static and a dynamic test mode for testing the input setup and hold time of memory devices. Depending on test requirements, we choose the static mode or dynamic mode. The static mode mainly targets to determine timing pass or fail by comparing the measured timing parameters and the required timing specifications. During the dynamic mode, the main goal is to measure timing margin variations under noisy conditions such as power fluctuations. The detailed circuit design will be following in the next section.

3.2 Prototype Chip Design

3.2.1 Circuit Architecture

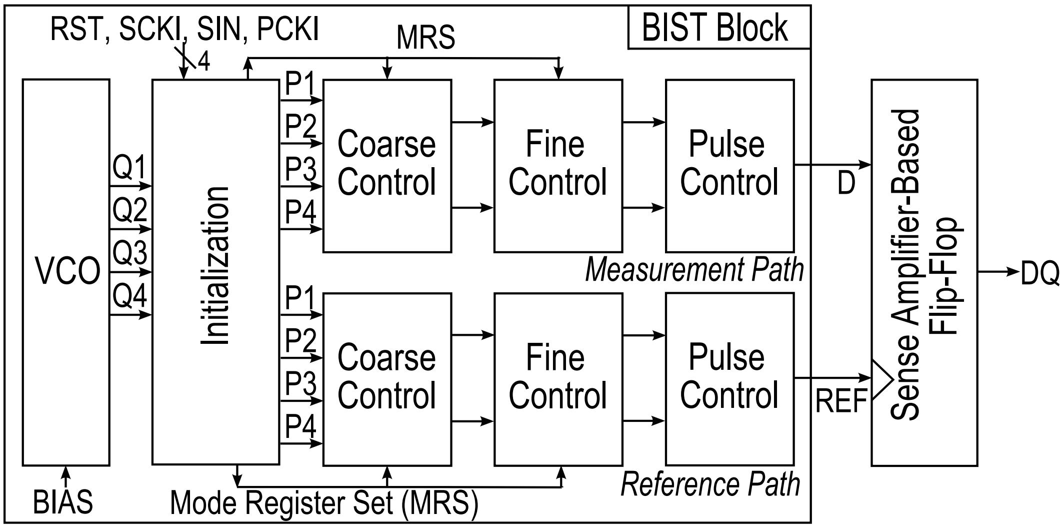
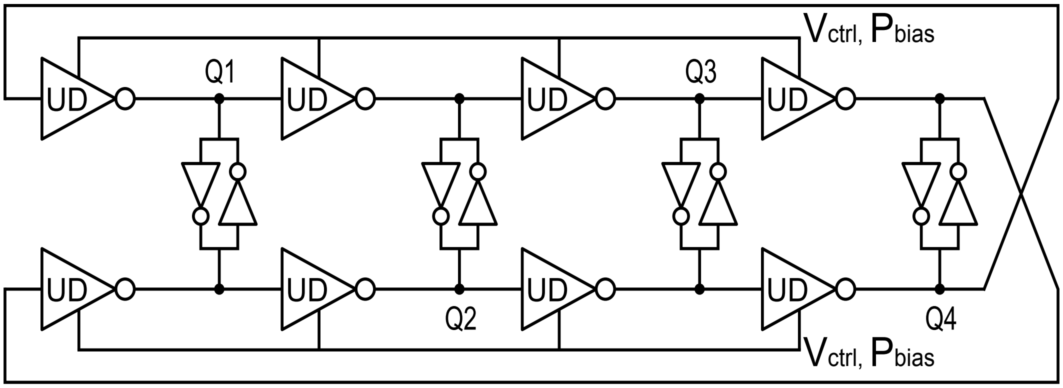


Figure 3.1: Chip block diagram

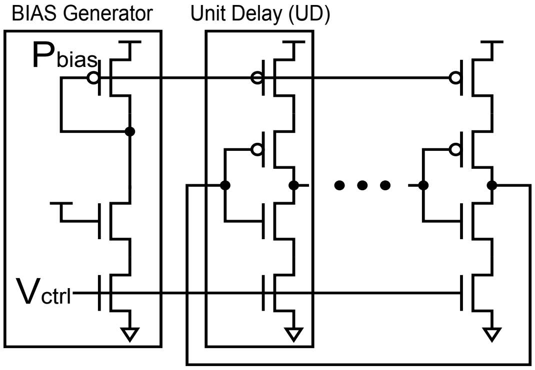
The fundamental goal of this test chip is to test high-speed interface signals using a low-cost test equipment. Therefore, I/O timing parameters need to be measured at low speed. Figure 3.1 shows the chip block diagram that we implemented. This test chip is composed of a BIST circuit and a sense amplifier based flip-flop (SAFF). This type of the SAFF is generally used for high performance I/O registers because of the accurate sampling characteristics [19]. The SAFF is designed to obtain test results as a multiple of the test clock period in addition to being used as a circuit under test.

The BIST scheme consists of signal generation, initialization, and edge-control circuits. The voltage controlled oscillator (VCO) generates signals 'Q1' to 'Q4'. The initialization block sets the initial values of internal circuitry. It includes serial-to-parallel logic to get external inputs with a smaller pin count. Using this logic, we need only four signals 'RST', 'SCKI', 'SIN' and 'PCKI' to set the initial conditions for testing. The signals can be set by a low-end tester because the input signals for testing do not need to be fast. After the test conditions are set, the data and clock are generated using the VCO based on-chip pattern generator. The edge-locations of the generated signals are controlled using the initialization, coarse control, fine control, and pulse control units. During the initialization, the mode register set (MRS) values are set by the tester to decide the control methods of the coarse and fine blocks. The 'D' and 'REF' shown in Figure 3.1 are the finally generated data and clock patterns, respectively. The MRS codes can be separately set for the data and clock paths of *'Measurement Path'* and *'Reference Path'*. Therefore, the patterns of 'D' and 'REF' are independently controlled. During the controls, the data and clock patterns are the inputs of the SAFF, and we test the setup and hold time of the SAFF by monitoring the output, 'DQ'. A certain time difference between two inputs is required to satisfy the setup and hold time of the SAFF. Accordingly, we initially set the relative time difference between two paths of *'Measurement Path'* and *'Reference Path'*. We test the SAFF while the time differences from zero to half clock period are controlled. The signal edges of 'REF' are precisely moved until the SAFF captures valid data during the test.

3.2.2 Signal Generator



(a) Differential type ring oscillator



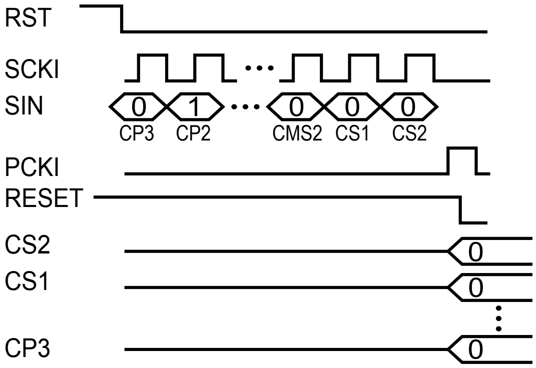
(b) Current starved unit delay cell

Figure 3.2: Phase generator

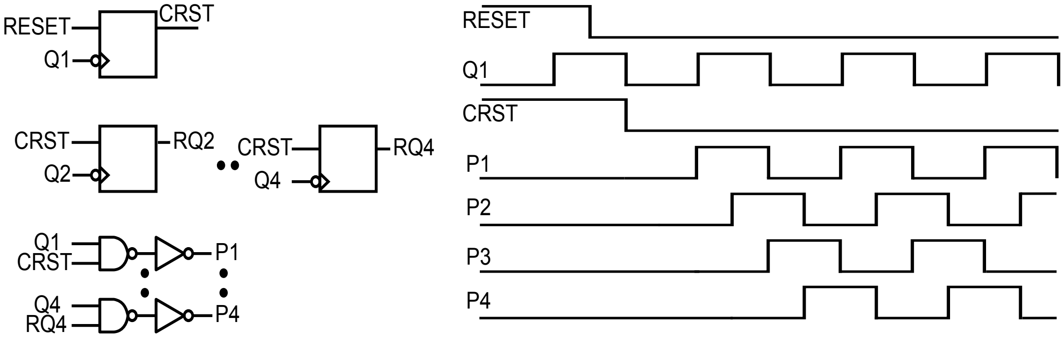
We implemented the embedded pattern generator using the VCO shown in Figure 3.2. The VCO produces the reference signals to generate data and clock patterns. The reference signals are the four phases of ‘Q1’, ‘Q2’, ‘Q3’ and ‘Q4’ coming from the VCO, shifted by 0 °, 90 °, 180 ° and 270 ° respectively. The VCO is structured with a pseudo-differential type, and consists of unit delay (UD) cells and feedback inverters as shown in Figure 3.2(a) [7]. The unit delay cell shown in Figure 3.2(b) is designed as an adjustable driver of a current-starved type [14] to calibrate the amount of current sources. The unit delay consists of current sources of PMOS and NMOS transistors, which are controlled by P_{bias} and V_{ctrl}, respectively. P_{bias} is controlled by the bias input, V_{ctrl}; the operating frequency range of the VCO can thus be managed by changing the control voltage. V_{ctrl} is connected to 'BIAS', an external input, as shown in Figure 3.1. The VCO produces octal phases of four differential pairs. We can use octal phases to increase the test resolution, but four phases are used as the source signals.

3.2.3 Initialization Unit

The BIST scheme produces both data and clock patterns, which are the inputs of I/O registers. The edge placements of two patterns are controlled to have the relative time difference. The amount of the relative time difference is programmable by giving different controls to data and clock paths. In order to satisfy the characteristics of both programmability and relative timing, the reference signals need an initialization process. During the initialization, the reference signals have a reset state to present the starting point of the signal sequences for controlling the edges. Additionally, MRS codes are set to determine the control methodology. Figure 3.3 shows the circuits and timings for the initialization procedure. In Figure 3.3, 'RST' is the chip reset. 'SIN' and 'SCKI' are the serial-input sequence and clock, which are used in the serial-to-parallel circuits. The incoming serial data of 'SIN' is synchronized



(a) Mode register set (MRS)



(b) Phase initialization (circuits and timing)

Figure 3.3: Initialization unit

and latched by 'SCKI', and the latched data is converted to parallel data. The parallel data is transferred to set the MRS codes. Therefore, the information of MRS codes is transferred through 'SIN' to the internal control logics. Once the sequence of 'SIN' is completed, the flag of 'PCKI' is toggled to indicate the end of the test setup.

The MRS codes determine data patterns, the amount of control, and test methods. The default values for each control are described in Table 3.1. The codes of 'Current Strength' change the driver strength of the unit delay

	Current Strength			Fine Control					
MRS	CP3	CP2	CP1	FS6	FS5	FS4	FS3	FS2	FS1
Default	0	1	0	0	0	0	0	0	0

	Fine Control						DATA	CLK
MRS	FMS6	FMS5	FMS4	FMS3	FMS2	FMS1	INV	INV
Default	1	0	0	0	0	0	0	0

	Self-Mode				Coarse Control			
MRS	RMUP	FMUP	RUP	FUP	CMS1	CMS2	CS2	CS1
Default	0	0	1	1	0	0	0	0

Table 3.1: Mode register codes for setting test sequences

cell; these control currents and thus calibrate the VCO operating frequency. The settings of 'CPx' determine the number of turn-on transistors to increase or decrease the current strength to trim the fine controls. The MRS codes of 'Coarse Control' and 'Fine Control' are used to control the amount of the timing interval. The 2-bit codes of 'Coarse Control' determine the initial time difference in addition to giving the test intervals of large delays. The 6-bit codes of 'Fine Control' are used to precisely control the time difference. Different controls are used for the two test modes : fix-mode and self-mode. The MRS codes shown in Table 3.1 are applied only for the fix-mode settings. In the self-mode, the fine control codes are automatically generated using internal counters. 'Self-Mode' equal to '0' indicates fix-mode. Also, the BIST scheme efficiently tests the DDR operations using various combinations of the MRS codes. For the testing, the 'RMUP', 'RUP', 'FMUP', and 'FUP' are defined in Table 3.1, where ‘R’, ‘F’, and ‘M’ denote the rising, falling, and measurement path, respectively. According to the codes, we set the fix-mode or the self-mode for the rising and falling edge-controls. Additionally, we have the function of ‘data inversion’ to generate various patterns for testing DDR operations. Depending on the ‘INV’ codes of the ‘DATA’ and ‘CLK’, we test ‘high-level (H)’ or ‘low-level (L)’ data. Therefore, we independently control the rising and falling edge, and thus analyze the setup and hold time depending on the data patterns. After all MRS codes are set, the ‘RESET’ is triggered by the toggle of ‘PCKI’. ‘RESET’ initiates the reference signals in addition to resetting the internal circuitry. Since the initial states at each internal node of VCO are nondeterministic, the initial phase is likely changeable at every power-up. Thus, any phases of ‘Q1’, ‘Q2’, ‘Q3’, and ‘Q4’ can be the first one, $0\,^{\circ}$. The issue increases the complexities of the edge-controls because our goal is to have controllability for the relative timing. Accordingly, the starting phase needs to be fixed to have a consistent ordering. Therefore, we implemented circuits such that ‘Q1’ is the first phase and all phases initially have low-level states. As shown in Figure 3.3(b), the ‘RESET’ is latched by ‘Q1’ to generate the trigger signal, ‘CRST’. Using ‘CRST’, the reset signals for the phases of ‘Q2’ to ‘Q4’ are generated. The phases of ‘P1’ to ‘P4’ are formed by the AND operations of the reset signals and the initial phases of ‘Q1’ to ‘Q4’. As a result, the four phases of ‘P1’ to ‘P4 are transferred to the timing edge-control logic.

3.2.4 Control Units

The generated four phases using the VCO and the initialization block are transferred to the control units. The control units consist of coarse, fine, and pulse control blocks. First, the timing edges of the phases are controlled by two steps of coarse and fine units. Then, the pulse control unit forms the final signal pattern using the rising and falling edge is controlled by two step control. The details of each block are as follows.

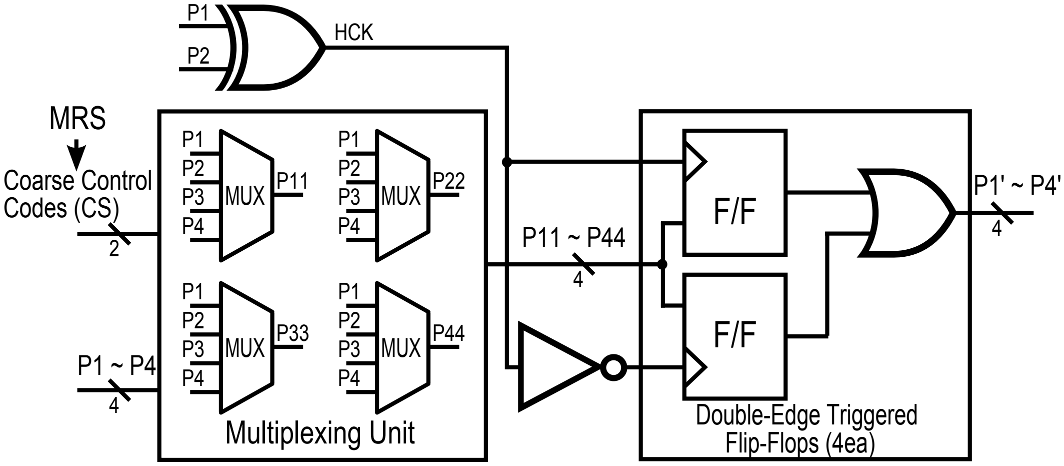


Figure 3.4: Coarse control unit

1) Coarse Unit: The coarse control unit selects one initial edge-location from four phases of ‘P1’, ‘P2’, ‘P3’ and ‘P4’. As shown in the block diagram of Figure 3.4, the coarse unit consists of a multiplexing unit and double-edge triggered flip-flops. The multiplexing unit consists of four multiplexors and the signal routings for the multiplexor controls to shift the phases with a certain phase sequence. Depending on the ‘CS1’ and ‘CS2’ of MRS codes,

MRS codes		Phase-shift results				Edge-location	
CS2	CS1	P1′	P2′	P3′	P4′	Rising edge	Falling edge
0	0	P1	P2	P3	P4	P1 to P2	P3 to P4
0	1	P1	P2	P2	P3	P1 to P2	P2 to P3
1	0	P2	P3	P3	P4	P2 to P3	P3 to P4
1	1	P1	P2	P4	P1	P1 to P2	P4 to P1

Table 3.2: Coarse control codes for phase shifting

the ‘P11’ to ‘P44’ are the shifted phases of the ‘P1’ to ‘P4’ coming from the VCO. However, glitches occur at every 90° during the phase shifting because the combinational logics are used. Therefore, a clocking is required to remove glitches. A double-edge triggered flip-flops (DEFF) are implemented for the clocking scheme. The DEFF decreases the speed overhead of the sampling clock because this flip-flop captures data at both rising and falling edge of the clock. Therefore, the sampling clock, ‘HCK, is simply formed by the XOR operation of ‘P1’ and ‘P2’ instead of creating an additional clock. Hence, the ‘P1′’ to ‘P4′ become the glitch-free signals for the final outputs of the coarse unit. The detailed operations for phase-shifting are described in Table 3.2. It shows the phase-shift operations considering both rising and falling edge for DDR operations. For instance, while the ‘CS2’ and ‘CS1’ are ‘L’ and ‘H’, respectively, the new signal edges of ‘P1′’, ‘P2′’, ‘P3′’, and ‘P4′’ are determined from ‘P1’, ‘P2’, ‘P2’, and ‘P3’, respectively. The operation indicates that the control ranges of the rising edges are between ‘P1’ to ‘P2’ and the falling edges are decided between ‘P2’ and ‘P3’. Similarly, using other MRS codes, the different control ranges are set. As a result, the timing edge-locations are selected with the resolution of 90 ° during the coarse control.

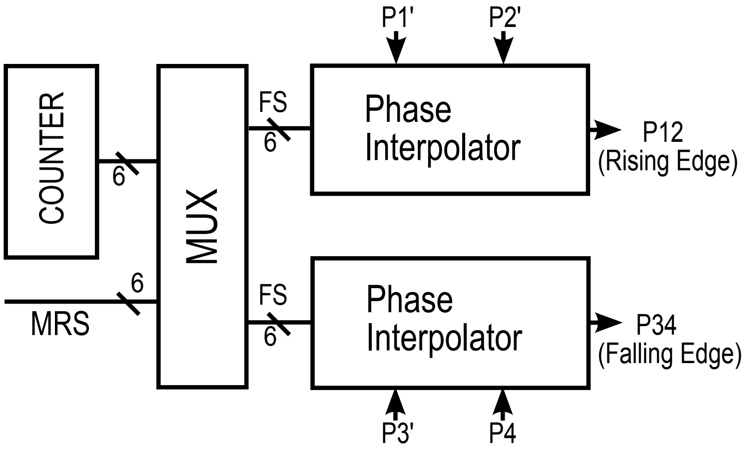


Figure 3.5: Fine control unit

2) Fine Unit: The fine control unit precisely controls the timing edges of signals coming from the coarse unit. The block is composed of counters, phase interpolators and multiplexors. The timing edges of ‘P1′’ to ‘P4′’ transferring from the coarse unit are the inputs of two phase interpolators as shown in Figure 3.5. The phase interpolator produces an interpolated phase between two input signals. The upper phase interpolator generates precise phases for the rising controls using ‘P1′’ and ‘P2′’. Similarly, the lower one operates for the falling controls using ‘P3′’ and ‘P4′’. The ‘P12’ and ‘P34’ are the outputs of phase interpolator and the interpolated results of two rising edges of ‘P1’ and ‘P2’, and ‘P3’ and ‘P4’, respectively. In order to have enough margin of one clock period while the rise and fall edges are controlled, we separately use two phase interpolators. Therefore, we independently control the rising and the falling edges of patterns. The 'P12' and 'P34', the output timing edges of PI, is determined by the current sources of the PI. The strength of the current sources is set by the fine control codes, and thus the weight of interpolating the timing edges can be controlled. In order to generate the fine control codes, we use MRS codes or internal counters. The multiplexors are used to select one of two test modes as shown in Figure 3.5.

The test resolution is determined by the coarse and fine controls. Since two inputs of the phase interpolators are associated with the signals coming from the VCO, the test resolution increases linearly with the VCO operating frequency. We can implement a separate clock generator such that the control units have a higher resolution regardless of the VCO operating frequency. However, it increases the hardware overhead. Moreover, the test resolution for both low-end and high-end system applications is not necessary to be identically high. A lower test resolution for low-end systems is acceptable because the required timing specifications are not very stringent. Moveover, since our BIST circuit is designed for a low-cost method, we do not use any additional clock generator to minimize the hardware overhead. Instead of using an additional clock generator, we control the test resolution of the generated signal patterns using the combinations of 'Fine Control' and 'Coarse Control'.

Using the coarse control unit, we initially choose one phase out of the four phases from the VCO. For the fine control unit, we implement 64 phase steps using the 6-bit code to control the current sources of the phase interpolators. The test resolution for the timing edges is thus determined by the number of bits of the fine control codes and the maximum time delay between the data and clock. Accordingly, the resolution is 'tCK/256' where tCK is the VCO operating frequency. For example, we achieve the resolution of 3.9 ps under the condition of 'tCK=1ns'. The resolution indicates the minimum time difference, and thus we can test the setup and the hold times of I/O registers with the resolution of 3.9 ps.

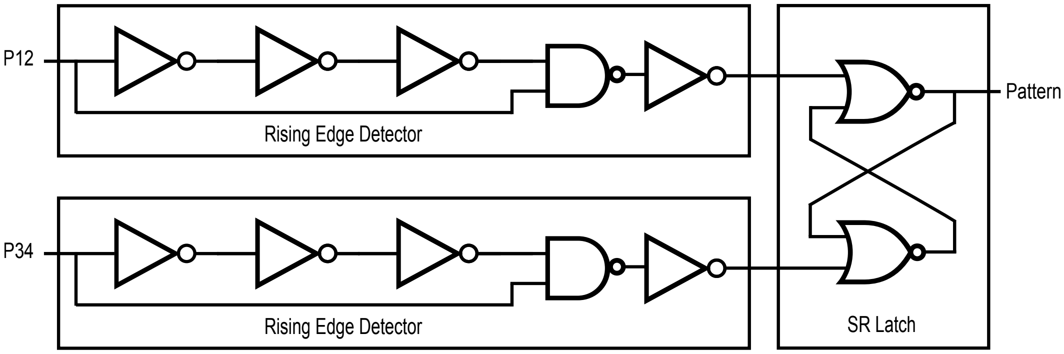


Figure 3.6: Pulse control unit

3) Other Circuits: After the two-level control, the pulse control unit shown in Figure 3.6 forms the final pattern using the rising edge detectors and SR latch. The edge detectors detect each rising edge of 'P12' and 'P34' coming from the fine control unit. The edge detectors are designed to avoid the overlap period between two inputs of SR latch. The SR latch forms the final patterns using the outputs of the edge detectors. As a result, the rising edges of 'P12' and 'P34' form the rising and the falling edges of the final patterns, respectively. The formed patterns are tested by the flip-flop to verify the BIST scheme as shown in Figure 3.1. We also implemented the sense-amplifier based flip-flop shown in Figure 3.7. The chip reset, ‘RST’, determines the initial logic state of the SAFF. After the test sequence of ‘SIN’ is completed and the coarse and fine controls are done, the SAFF holds the captured data. On the other hand, we can reset the output of the SAFF using ‘RST’ depending on the measurement methods. The outputs of the SAFF are analyzed to validate the test results. We evaluate pass or fail by monitoring the output, ‘DQ’, and also measure the required setup and hold time to capture correct data.

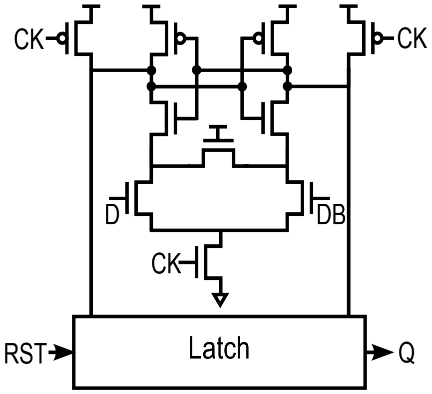


Figure 3.7: Sense-amplifier based flip-flop

3.2.5 Timing Operations

The timing operations of coarse and fine control blocks are described in this section. All the timing operations describes based on the test operations of the fix-mode, in which all control codes are fixed to the same values at every cycle. Figure 3.8 presents the timing diagrams of the coarse unit according to the coarse control codes (CS). It shows the cases of ‘CS=00’ and ‘CS=01’ while all the fine control codes are set to ‘000000’. As can be seen in Figure 3.8, since we set all the fine control codes to '000000', the patterns of 'D' are formed by the rising edges of 'P1'. Thus, the pulse widths and the falling edge-locations of 'D' can be varied using the coarse control codes. Figure 3.9 shows the fine control operations depending on different fine codes. In order to analyze the operations, the coarse control codes are fixed to '00'. The fine control codes are separately set for both the rising and falling edges. In Figure 3.9, the 'FS@R' and 'FS@F' indicate the fine control codes for the rising and the falling edges, respectively. In case of 'FS@R=100000' and 'FS@F=000000', the rising edge of 'D' is determined at the middle point between the rising edges of 'P1′' and 'P2′'. The falling edge of 'D' is located at the rising edge of 'P3′'. Similarly, in case of 'FS@R=000000' and 'FS@F=100000', the rising edge of 'D' is located at the rising edge of 'P1′'. The falling edge of 'D' is located at the middle point between two rising edges of 'P3′' and 'P4′'. As a result, the coarse unit

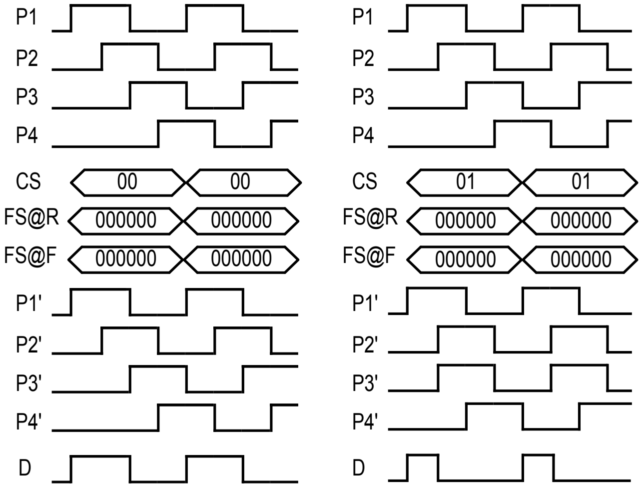


Figure 3.8: Coarse control unit operation

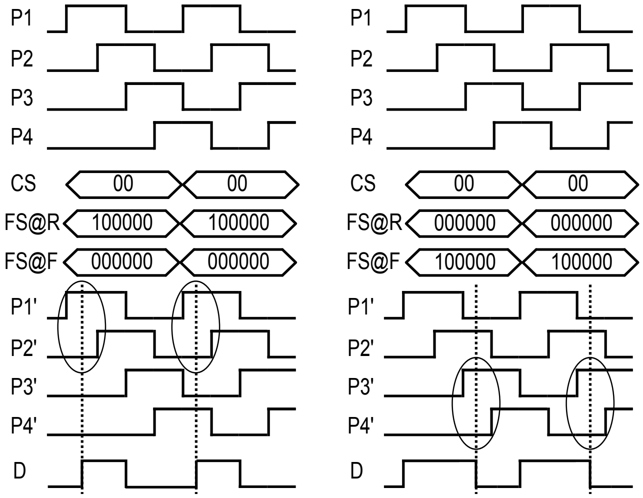


Figure 3.9: Fine control unit operation

decides the initial timing edge-location and pulse width. Starting from the initial edge-location, the fine control unit precisely moves the timing edges.

Figure 3.10 describes the overall timing diagram including the SAFF operations. The 'D' and 'REF' are the generated patterns from the measurement and the reference paths, respectively. The 'D' and 'REF' are the data and sampling clock of the SAFF and the 'DQ' is the latched result. The BIST operations are verified by monitoring the 'DQ' while the relative timing difference between data and clock is calibrated. We set 'CS' to '00' for both the measurement and the reference paths such that the same coarse code for two paths creates the maximum time difference of '0.25 tCK'. If we expect the required timing margin to be less than '0.25 tCK', we can set the same coarse code to generate the data and clock patterns. If we can roughly estimate the required timing margin, we initially set a smaller time difference to reduce the

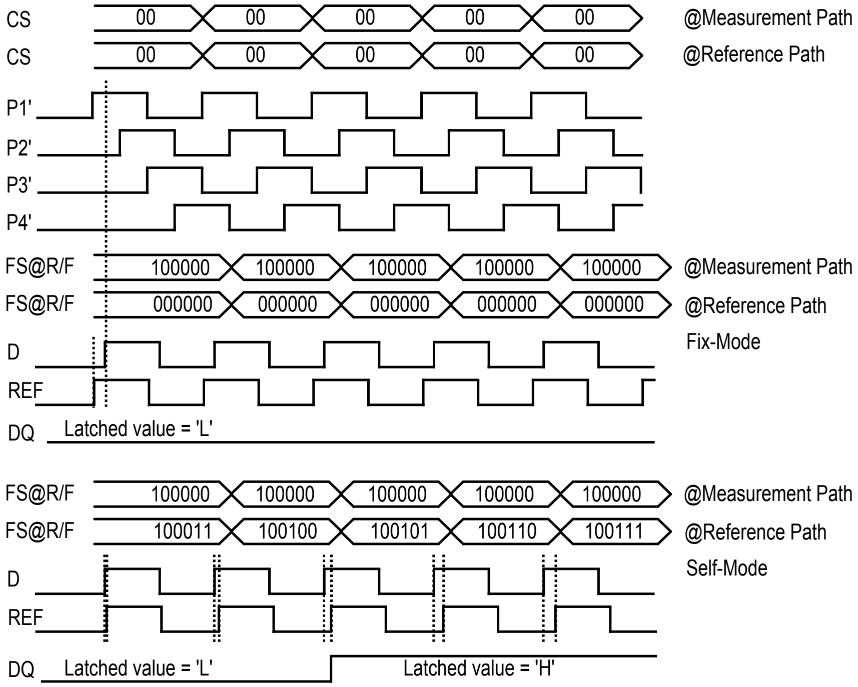


Figure 3.10: Overall timing diagram

testing time. Additionally, the same control code for both the rising and falling edge is asserted to generate patterns with 50% duty cycle. The fine control codes are separately generated for two paths to precisely control the relative timing difference. Figure 3.10 also explains the fix-mode and the self-mode tests for the fine control unit. To apply the fix mode for both the measurement and reference paths, we set ‘FS’ to ‘100000’ and ‘000000’, respectively. Thus, the code difference of ‘D’ and ‘REF’ is ‘100000’ which indicates the time delay of ‘0.125 tCK’. If the ‘0.125 tCK’ is enough to satisfy the setup time of ‘D’ with respect to ‘REF’, the ‘DQ’ captures the valid ‘L’ data. At every test trial, we change the MRS code to gradually decrease the time difference and monitor ‘DQ’ at a time to check timing pass or fail. By reading out the MRS codes, we measure the required setup and hold time, where the MRS codes indicate the number of the required tCK cycles ($\mathrm{N_{code}}$).Therefore, the required timing margin ($\mathrm{t_{margin}}$) is calculated by Equation (3.1) where $\mathrm{t_{RES}}$ indicates the resolution of phase interpolator.

$$\mathrm{t_{margin}} = \mathrm{N_{code}} \times \mathrm{t_{RES}} \tag{3.1}$$

On the other hand, during the self-mode, the fine control codes automatically increase or decrease at every cycle using the internal counters. The two test modes are selectively used for two paths to generate the relative time difference at every tCK cycle. In Figure 3.10, unlike the measurement path, the self mode is assigned to the reference path. Accordingly, the fine control codes are changed at every tCK cycle to control the edges of ‘REF’. Therefore, whenever ‘D’ is captured at the rising edge of ‘REF’, the timing margin is gradually decreased. If the expected data is ‘H’, the ‘DQ’ is held low during the first several cycles and then transits from ‘L’ to ‘H’ when there is enough margin to capture ‘H’. The self-mode is thus effective to reduce the test time and measure the amount of variations using the cycle-by-cycle control method.

3.2.6 Data Pattern Generation

One of features of this technique is to effectively test DDR operations of memory interfaces. For testing DDR operations, we consider that ‘L’ and ‘H’ data of ‘D’ are latched at both the rising and the falling edge of ‘REF’ as shown in Figure 3.10. Instead of sampling data during the whole clock period, we define the time difference between ‘D’ and ‘REF’ using the coarse control codes and sample data only within this region. Additionally, we effectively test DDR operations by combining the ‘INV’ mode shown in Table 3.1. The ‘INV’ mode chooses the data inversion of ‘D’ and ‘REF’. Therefore, the ‘D’ and ‘REF’ generate the four patterns of ‘HH’, ‘LH’, ‘HL’, and ‘LL’, where ‘H’ and ‘L’ indicate the inverted and non-inverted data. We simply implemented the inversion by using an inverter gate because it is enough to show the application of the BIST circuit. For the real I/O circuits, a phase mixer instead of a simple inverter is usually used to minimize duty distortions because the duty ratio critically affects the interface timings of DDR memory. On the other hand, we also show how the BIST scheme is applied to test DDR operations by simply modeling the I/O registers. Moreover, in order to increase observability, we control independently the rising and falling edge by changing the codes of ‘RUP’ and ‘FUP’. Once the starting point of the controls is defined, we test the setup time of ‘D’ with respect to the rising edge of ‘REF’. If we set ‘RUP’ to ‘H’ and ‘FUP’ to ‘L’, the hold time margin between the rising edge of ‘D’ and the falling edge of ‘REF’ is enough to capture data correctly. Accordingly, while we test the setup time, we do not need to consider the timing failure due to the hold time violation. Similarly, we can set ‘RUP’ to ‘L’ and ‘FUP’ to ‘H’. The hold time margin between the rising edge of ‘D’ and the falling edge of ‘REF’ is gradually decreased until the setup time margin is met. Therefore, we test individually the setup and hold time. Furthermore, the BIST scheme achieves the adjustable pulse width of patterns as shown in Figure 3.8 and

Figure 3.9. The feature increases the selectivity of the starting points to control the rising and falling edge and decrease the testing time. It also increases the testability of duty cycle distortion. If we use the same codes for both the rising and falling edge, it excludes the effects caused by duty cycle distortion. On the other hand, the different code settings to two paths generate the reference signals with duty cycle distortion. Thus, we measure the effect of duty cycle distortion on the interface timing.

3.2.7 Low-Cost Measurement Methods

This section introduces a low-cost method which can measure the test results using a low-cost ATE in addition to having a low-area overhead. The BIST operations do not need to have high-speed external signals from a tester to test the specification such as the setup and hold time. This is because the external inputs are used only to set the MRS codes. The external outputs for measuring the test results also do not require a high-speed tester because of the presented measurement methods. The BIST scheme provides two methods to increase observability in measuring the test results. First, we measure the number of the required tCK cycles of 'REF' where the SAFF has enough timing margin to latch the expected data. The counters update the fine control codes at every tCK cycle. The updated codes change the time difference between 'D' and 'REF'. When the timing interval meets the required timing margin and the data value of 'DQ' is changed, we read out the counter values to measure the number of tCK cycles. We calculate the required setup and hold time using the code difference between the reference and measurement paths. Accordingly, we calculate the number of tCK cycles for the timing parameters using Equation (3.1).

We measure the pulse width of ‘DQ’ (t_{PW}), the output obtained from the BIST circuit. As shown in Figure 3.10, the first transition point of ‘DQ’ is changed depending on the time difference of two inputs. Next, we reset the internal node of the SAFF for the second transition in order to make ’DQ’ a pulse. The second transition of ‘H’ to ‘L’ is asserted at the same point regardless of the timing interval between ‘D’ and ‘REF’. In order to have the second transition at the same point, ‘RESET’ is latched by the chip reset signal, ‘RST’, which is synchronized with ‘SCKI’. The ‘RESET’ is also latched using the VCO clock to form the pulses based on the tCK cycle. Therefore, the synchronized ‘RESET’ initializes the internal node of the SAFF. The reset state is set to the opposite value of the expected data to make the pulses. For instance, in case of capturing ‘L’ data of ‘D’ at the rising edge of ‘REF’, the ‘DQ’ transits ‘H’ to ‘L’ at first. By the assertion of ‘RESET’, the second transition of ‘L’ to ‘H’ is performed. As a result, we obtain the different pulse widths according to the required timing margins. By comparing the pulse widths, we measure the amount of the timing violations. Moreover, we can extend the scheme to measure the amount of variations caused by power noise, jitter, crosstalk, process, voltage, and temperature.

Figure 3.11 shows the block diagrams measuring the pulse widths by using the ‘RESET’ scheme. The structure of Figure 3.11 can be considered

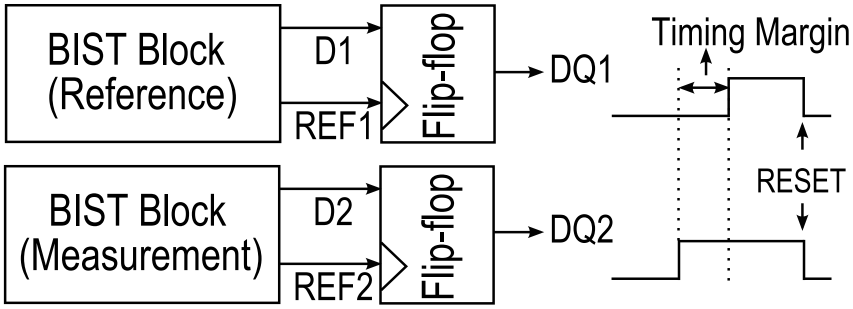


Figure 3.11: A measurement method

to test the amount of setup and hold time violations. The upper block produces the reference pulse. The measured pulse width from the lower block is compared with the reference one. The timing of the reference block is set by the given specification of the setup and the hold times [1]. In the case of DDR-667 [1], the specification of the setup and the hold time for data paths are 175ps and 100ps, respectively. Accordingly, we use the fix-mode in the reference block to set the time difference between 'D1' and 'REF1' based on the required specification. For the measurement block, we use the self-mode and calibrate the edges of 'REF2' at every tCK cycle until the SAFF has enough margin. As a result, the 'DQ1' and 'DQ2' pulses are generated. If the pulse width of 'DQ2' is wider than 'DQ1', it means that the measurement path satisfies the required timing specification. The difference of the pulse widths indicate the amount of over-specification. On the other hand, because of mismatch, the 'REF2' can capture data later than 'REF1'. It means that a timing violation has happened in the measurement block. The difference of the pulse widths indicate the amount of violated specification. For I/O circuits, the factors such as physical mismatch, power supply fluctuations and crosstalk noises affect the timing margin. The proposed BIST scheme is thus applied to measure timing variations due to these factors. Even if we do not have the additional upper block to generate the reference pulse, we still measure the timing variations through test iterations only using the measurement block. Therefore, once we measure the pulse widths, the difference between them shows the amount of over-specification, violated specification, and timing variations depending on the BIST structure and control methods. Since the pulse width is formed based on the tCK cycle, the measurement results are quantified by converting the pulse width to the time unit. One tCK period thus indicates the resolution of the phase interpolator. Therefore, the measurement result for setup and hold time, $\mathrm{t_{param}}$, is calculated as shown in Equation (3.2).

$$\mathrm{t_{param}} = (\Delta\ \mathrm{t_{PW}} \div \mathrm{tCK}) \times \mathrm{t_{RES}} \tag{3.2}$$

As a result, the BIST scheme does not need any high-speed signal for testing the I/O timings. Using the proposed measurement methods, the scheme is compatible with a low-cost ATE. Furthermore, the external signals for testing are operated in the BIST scheme without critical performance degradations under the conditions of large capacitive loadings. Therefore, the BIST method is also available for the I/O timing measurements during the wafer-level test.

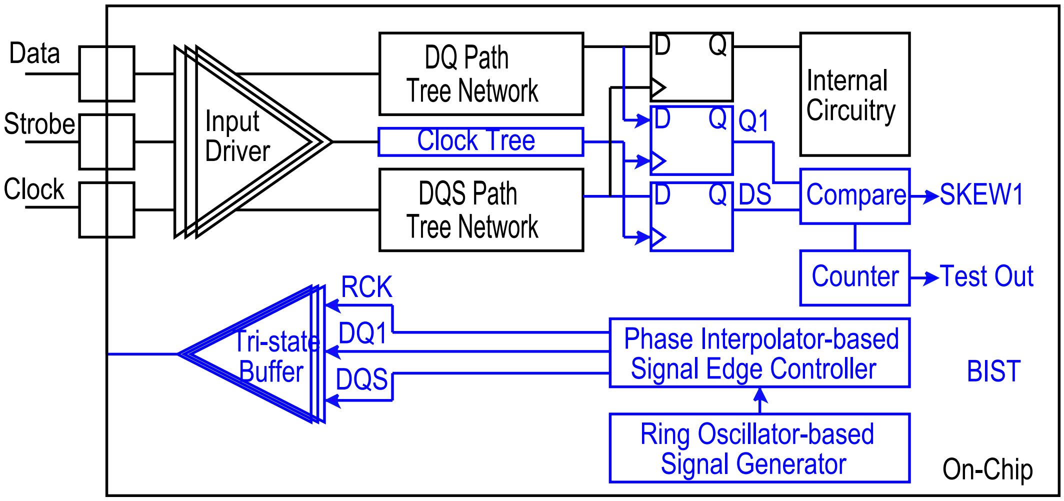


Figure 3.12: Configuration for testing setup and hold times

3.3 Measurement Results

Figure 3.12 shows the test chip configuration for testing the setup and hold time. This BIST circuit has been implemented and fabricated with a 0.18-μm TSMC one-poly six-metal CMOS technology (CL018). To validate the BIST circuit, we also designed the CUT including input drivers and DQ/DQS path trees. Using the BIST circuit, we generate timings of DQ and DQS, and measure the delay mismatches between DQ and DQS occurred in the CUT. The BIST circuit consists of a signal generator and a signal-edge controller. The ring oscillator based signal generator generates DQ, DQS and reference clock (RCK) signals for test inputs. Using the phase interpolator based signal-edge controller, we detect each transition timing of DQ and DQS with respect to RCK to measure the time difference between DQ and DQS. To find the transition timings with high resolution, the signal edges need to be controlled with a small time delay. Using the ring oscillator and the phase interpolator, two-level coarse and fine control circuits are implemented, respectively. The RCK, DQ and DQS timing-edge locations are digitally changed using the two-level controls. Thus, one test clock period and the total delay range are determined by the frequency of the ring oscillator that is controllable by changing a supply voltage or a current strength.The minimum time delay is set by the resolution of the phase interpolator (t_{PI}). The BIST circuit performs both the detection of the transition timings and the measurement of the time difference at the same time. Thus, the test results are quantified as a multiple of the test clock cycle, which indicates a slow-speed signal. Figure 3.13 explains the cycle-by-cycle edge control method and slow-speed test results. The BIST circuit changes the timing edges of DQ and DQS with a resolution of 20 ps at every test clock cycle to find the transition timing. The RCK samples DQ and DQS, and the sampling results (Q1 and DS1) are compared. Thus, the time difference between DQ and DQS, SKEW1, results in a multiple of the

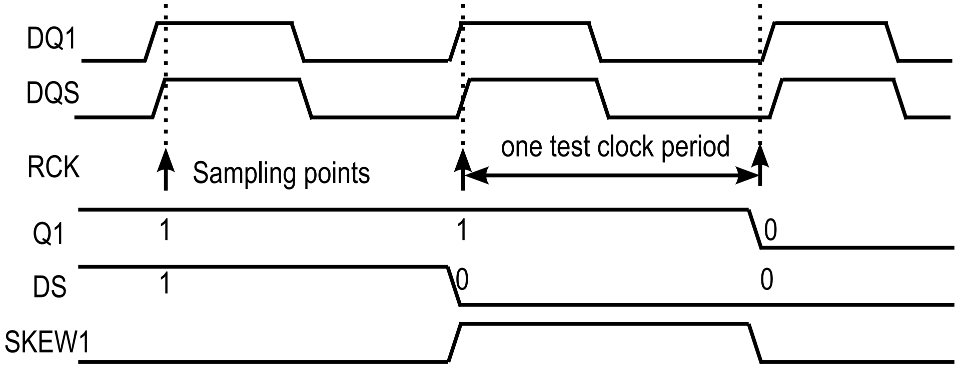
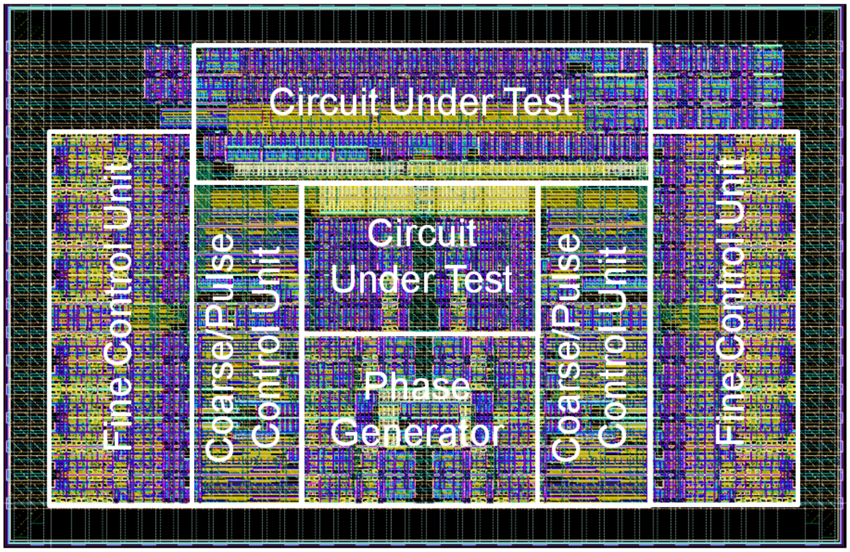
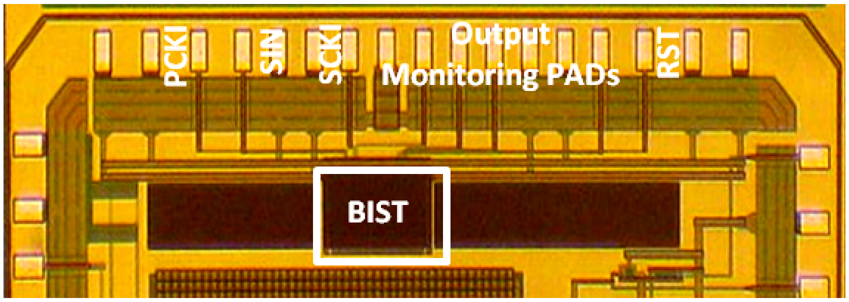


Figure 3.13: Detect transition timings to measure time differences

test clock period where one test clock period is corresponding to 20 ps (the resolution of the phase interpolator). Therefore, the BIST circuit only requires to count the number of the test clock cycle to detect the delay mismatches and thus measure I/O timing parameters.



(a) Layout



(b) Die photo

Figure 3.14: Chip layout and die photo of BIST circuit

Figure 3.14 shows the full chip layout and the die photo of the BIST

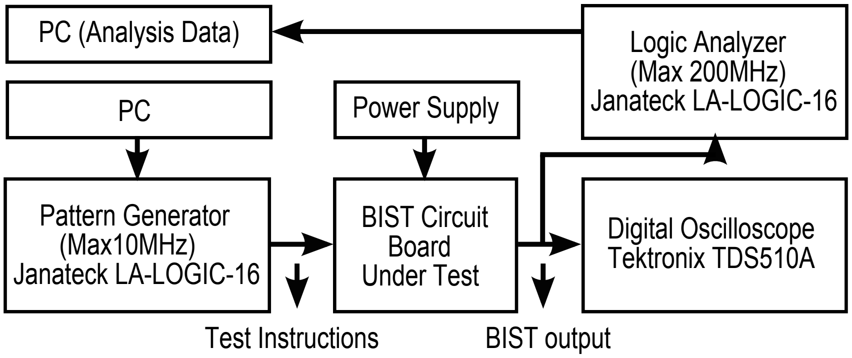
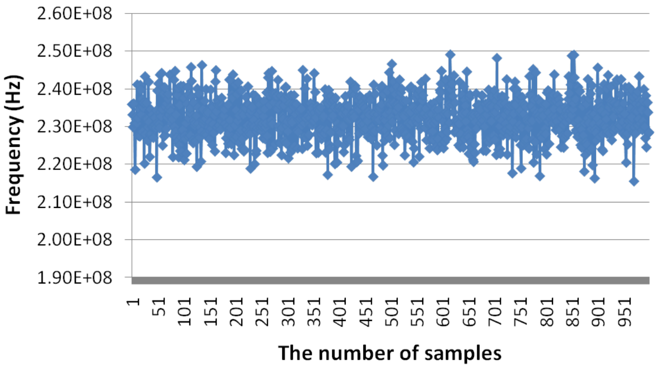


Figure 3.15: Measurement setup

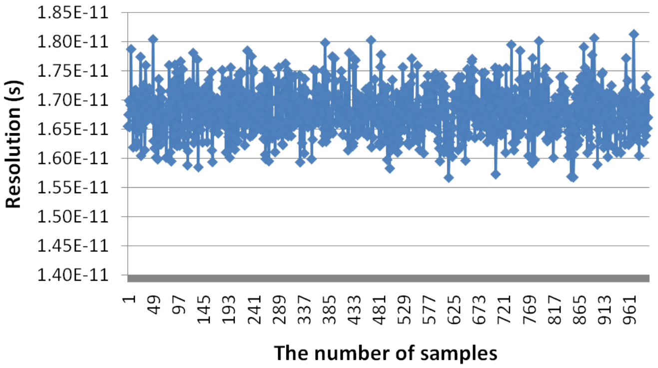
scheme. The total chip size is *'280μm × 130μm'* excluding bond pads and ESD circuitry. The measurement setup for testing is structured as shown in Figure 3.15. The PC-based Janateck LA-Logic was substituted for the low-cost ATE. It operates as a pattern generator to give the test instructions to the BIST circuit board. The equipment provides the input sequences of 'RST', 'SCKI', 'SIN', and 'PCKI' depending on the test modes. It is also used as a logic analyzer to analyze the latched values of 'DQ', coming from the BIST circuit. The 'DQ' is also monitored using the oscilloscope. The BIST scheme has been validated in the static and the dynamic mode for testing the setup and hold time.

3.3.1 Signal Generation

The VCO and phase interpolator have been implemented to generate the reference signals and control the signal edges. Monte-Carlo simulations are performed to verify the basic functions and estimate the VCO operating frequency ranges. The statistical parameters are given by Hspice model of the



(a) VCO operating frequency



(b) The resolution of phase interpolator

Figure 3.16: Monte-Carlo simulation results

0.18-μm TSMC CMOS process. We performed the post-layout simulations by setting both 'BIAS' and 'VDD' to 1.8 V, where 'BIAS' and 'VDD' indicate the bias control voltage, 'V_{ctrl}', of the VCO and the power supply voltage, respectively. Figure 3.16 shows the results that the average VCO operating frequency is 230 MHz and the average resolution of phase interpolator is 17 ps. The VCO operating frequencies were also measured by probing the output monitoring pad on the BIST chip. The comparison results of the simulation and the measurement are shown in Table 3.3. The measurement result of the VCO operating frequency, tCK, is 210 MHz with 'VDD' and 'BIAS' set to 1.8 V, which is less than 230 MHz obtained from the simulation. This difference is because of the process variations and the additional capacitive loadings of the testing environment. However, the results are fairly good because the VCO is used as the sampling clock in the BIST scheme and we count the number of cycles to quantify the measurement results. Therefore, the VCO does not need to guarantee an accurate value of frequency. Based on the measurements of the VCO operating frequency, we test the setup and hold time in the static and the dynamic-mode. The measurement results are discussed in the next section.

BIAS	SIM.	MEAS.
0.9V	145MHz	110MHz
1.2V	175MHz	145MHz
1.5V	205MHz	180MHz
1.8V	230MHz	210MHz

Table 3.3: Reference clock operating frequency

3.3.2 Static-mode Test

The test modes are defined based on how the relative timings between two paths of the measurement and the reference are controlled. The relative timings are changed with the fixed value in the static-mode and the variable values in the dynamic-mode at each tCK cycle. The switch between the two modes is only applied to set the fine control codes. In order to perform the functions, one of two paths is always operated in the fix-mode. All the control codes for the measurement path are generated using the fix-mode. On the other hand, the timing controls of the reference path are operated in the fix-mode or the self-mode depending on the test mode. In the static-mode test, the fine control codes of the reference path are fixed irrespective of the tCK cycle. Additionally, the codes are set to different values to change the relative time difference at each test iteration. The MRS code settings for the static-mode test are as follows. As shown in Figure 3.10, basically, we used the same control codes for both the rising and the falling edges to produce signals with 50% duty cycle. Hence, regardless of the test modes, we set 'CS' to '00' for the coarse controls of both the measurement and the reference paths. Also, the 'RMUP', 'FMUP', 'RUP' and 'FUP' are all set to '0' to define the fix-mode. Thus, the fine control codes have the same value at every tCK cycle. For the measurement path using the fix-mode, we fix 'FS' to '100000' for the fine control codes. On the other hand, the 'FS' codes of the reference path are increased by one at every test trial starting from '100000'. During the procedure, we monitor the output results of SAFF and read out the code values.

The SAFF can capture data when 'D' and 'REF' are aligned under the ideal conditions of the zero setup and hold time, and perfect delay matching between 'D' and 'REF'. Therefore, we expect that '0.25 tCK' is enough setup and hold time margins for the SAFF to correctly capture data. Based on

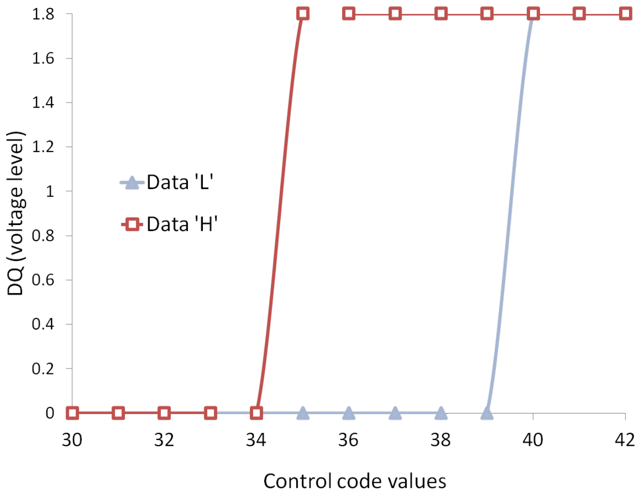
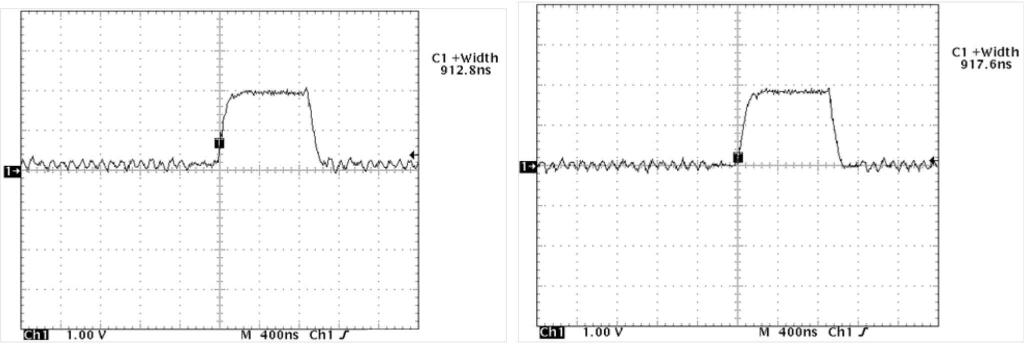


Figure 3.17: Test results in the static mode

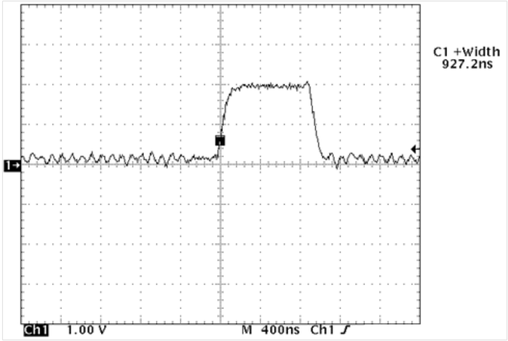
the estimated margin, we determine the testing range. Accordingly, the BIST block is tested in the fine control-code range of ‘011110’ to ‘101010’. Between this code range, the experimental results are obtained as shown in Figure 3.17. The x-axis represents the fine control codes of the reference path and the y-axis represents the SAFF output voltage-level of ‘H’ or ‘L’. The measurement results include the cases of testing the data patterns of ‘H’ and ‘L’ for ‘D’, data input. Also, the results comprise the test vectors using the inversion codes of ‘DATA’ shown in Table 3.1. Accordingly, the SAFF captures the ‘H’ and ‘L’ data at the rising edge of ‘REF’. If setup and hold times are zero, the edges of ‘D’ and ‘REF’ will be aligned for ‘100000’. However, because of nonzero value of setup and hold time, the SAFF captures the ‘H’ and ‘L’ data of ‘D’ at the fine control codes of ‘100010’ and ‘100111’, respectively. The difference between the ideal and the measurement codes corresponds to the required setup and hold time. The code differences for ‘H’ and ‘L’ data are ‘2’ and ‘9’, respectively. We quantify the codes by converting to the time unit using Equation (3.1). However, this chapter does not cover the method to guarantee the resolution of the phase interpolator. Therefore, we estimate the resolution of the phase interpolator based on the simulation results shown in Figure 3.16 and the measurement results of the VCO. As shown in Table 3.3, we obtained the VCO operating frequency of 210 MHz. Hence, we estimate that the resolution of the phase interpolator is ‘20 ps’ because one clock period has 256 phase steps. Therefore, the setup and hold times are ‘40 ps’ and ‘180 ps’ for ‘H’ and ‘L’ data, respectively.

3.3.3 Dynamic-mode Test

The dynamic-mode test implements the variable relative timings at every tCK cycle. The reference path is set to the self-mode in the dynamic-mode test. The control codes are set as follows. For the coarse controls, the same codes with the static-mode test are set. In order to define the self-mode only for the reference path, the ‘RMUP’ and ‘FMUP’ are set to ‘0’ but the ‘RUP’ and ‘FUP’ are set to ‘1’. For the fine controls, we set ‘FS’ to ‘100000’ for the measurement path. On the other hand, the fine control codes of the reference path are automatically changed at every tCK cycle by using the internal counters. The dynamic-mode test was performed using the measurement method shown in Figure 3.11. Using the dynamic-mode test, we detect the amount of margin degradations due to variations such as power



(a) pulse width=912.8 ns (b) pulse width=917.6 ns



(c) pulse width=927.2 ns

Figure 3.18: Test results in the dynamic mode

supply and ground noises. Such factors affect the time differences between 'D' and 'REF' and thus cause the timing margin variations. Figure 3.18 shows the timing variations by monitoring the pulse waveforms of 'DQ'. As shown in Figure 3.18, we obtained the pulse width variations of 912.8 ns, 917.6 ns, and 927.2 ns with operating frequency of 210 MHz (tCK = 4.8 ns). Each pulse width is a multiple of tCK. Through iterative testing, these three cases of pulse widths are obtained. The maximum difference of the pulse widths is 14.4 ns. The difference of 14.4 ns indicates the timing variation of '3 tCK cycles', where tCK is 4.8 ns. So, the timing variation is 60 ps using Equation (3.2). Thus, we measure the setup and the hold times using the static-mode and also detect their variation using the dynamic-mode.

3.4 Conclusion

This chapter presents a BIST scheme including the phase interpolator based on-chip timing sampler to implement a low-cost measurement technique. The BIST circuitry includes the measurement circuitry compatible with low-cost testers. The measurement methods of the static and dynamic modes were presented to increase testability and observability. From the experimental measurements, we obtained the setup and hold time of 180 ps and the timing variation of 60 ps. The measurement results are obtained by simply counting the number of clock cycles, where one clock cycle indicates 20ps. We measure the timing specifications without using any high-speed signals from the tester. Therefore, the BIST scheme is available to test the I/O performances during the wafer-level test and can also be applied in a Soc environment including high-speed memory. In viewpoints of area overhead, since the pattern generator and control blocks can be shared among parallel I/Os, the area overhead is negligible compared to the overall memory chip size. This chapter shows measurement results for the input setup and hold time but this approach can be extended to test other I/O timing specifications.

Chapter 4

BIST Solution for DDR Memory Output Timing Test and Measurement

4.1 Introduction

The previous chapter described the low-cost timing measurement method using the phase interpolator based on-chip timing sampler for high-speed memory test applications. In this chapter, we apply the low-cost measurement method for testing DDR memory output timing parameters and present the measurement results.

The DDR interface transfers two data words per clock cycle. The DDR operation critically affects output timing parameters because of data skews and the rising and falling slew rate mismatches. As described in [67], the output timing specifications of memory are the most stringent and affected easily by the interconnection and calibration condition of the probe card. With the increasing data rates, at-speed testing for DDR output timing is more challenging due to the tester clock frequency and thus on-chip method is required to support at-speed DDR timing test. Moreover, it is crucial to measure output timing parameters with high accuracy without increasing system and test cost. The key challenges for testing high-speed parallel source synchronous output timings are as follows:

- Timing validation between data and data strobe
- Phase alignment between data and data strobe
- Per-pin skew measurement
- The requirements of high accuracy and resolution

In order to satisfy the above requirements, a BIST circuit is a promising technique to decrease test cost and development time while providing good test accuracy and resolution. Previously, BIST architectures for testing high speed memories have been developed [8, 58]. In [58], a BIST method to support at-speed testing for high speed DDR memory was presented. These methods improve testability, decrease test time, and dispense with expensive external test equipment. However, most previous work focus on testing memory cores [54] and little research has been done on testing DDR memory interface timing. In [20, 67, 69], DDR timing optimization and characterizations were explored but they did not cover the output timing measurement methods. One of the on-chip test methods is to adjust the strobe delay using a digitally controlled delay-line such that the strobe edges are located in the middle of the data [9]. However, this work also does not cover the parametric timing tests. In [70], a vernier delay line based time-to-digital converter (TDC) was presented to generate time delay between data and data-strobe for testing memory output timings. However, the delay line needs to be very long to measure a long period of time, which increases area overhead. Thus, this TDC method is not very effective for measuring I/O timing in memory devices that require to operate over a wide frequency range. Moreover, this method achieves a high resolution of 10 ps, but requires a large hardware overhead of '1.62 mm × 2.36 mm'. Moreover, since the technique is based on high performance ATE, it is costly too. A timing generator similar to TDC was implemented using DLL for testing the memory output in [13]. However, these methods need an external high frequency clock and thus are not compatible with low cost testers.

This chapter presents a low cost BIST scheme to support at-speed testing and measurement for DDR memory output timing parameters [34]. This method generates a relative time delay between data and clock, which does not need any external clock. While a time delay is precisely controlled, timing margins are measured using low-cost testers because high speed signals are not necessary for testing. Moreover, this chapter extends the methods described in Chapter 3 to increase testability for a higher speed timing operation. Since output timings are more stringent than input timings, we present a scheme to resolve the frequency limitation of the on-chip timing sampler. This chapter also presents a novel method for testing the resolution of the phase interpolator, the key circuit of the fine control unit of the on-chip timing sampler. Moreover, a novel self-diagnosis scheme is implemented by adding counters to the output of the BIST scheme. Since test results come out as a multiple of test clock cycles, this scheme can give information of timing pass/fail and margin variations to the tester by simply reading the counter values. Furthermore, this chapter shows chip measurement results for output timing variations due to per-pin skews, slew-rate change and switching noise.

4.2 Design Methodology

4.2.1 Design Background

Figure 4.1 shows a basic memory output timing diagram with clock (CLK, /CLK), data-strobe (DQS, /DQS) and data (DQ). The memory output timing specifications for t_{DQSCK} and t_{DQSQ} are also shown in Figure 4.1. t_{DQSCK} describes the allowed range for rising and falling DQS edges with respect to CLK. t_{DQSQ} indicates the allowed skew range between DQS and its associated DQ signals. t_{DQSQ} is the most stringent timing parameter with 100 ps for DDR3-1600 (1.6 GHz) [2]. Output timing variations (skews) appear due to process, changes of operating conditions (supply voltage, temperature etc.), slew-rate changes, simultaneous switching output (SSO) noise [25], and pin-to-pin mismatch. Furthermore, memory devices need to align DQ and DQS timings with respect to CLK for the output paths. A DLL generates the synchronized clock for the alignment, and thus the timing skews between DLL

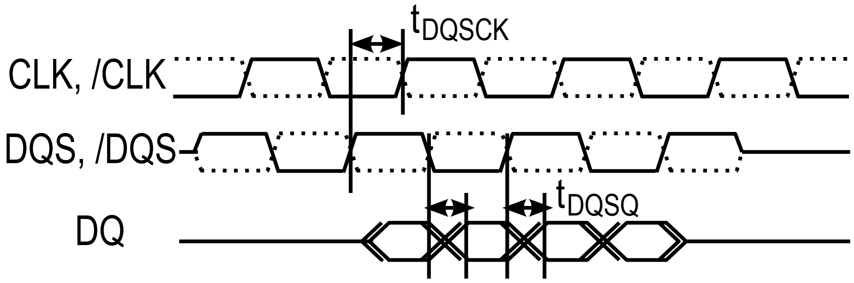


Figure 4.1: tDQSQ output timing specification

clock and DQ/DQS data also affect the output timing specification. This chapter also presents a method to measure output timing variations due to slew-rate change and pin-to-pin mismatch related to SSO noise. The SSO noise is associated with switching voltage, switching frequency, slew rate, mutual inductance and interconnection capacitance [29]. It critically affects propagation delay and signal integrity, especially in systems with data rates over a few hundred MHz. In memory systems with many DQ pins, output timing performance changes with the number of DQ bits switching at a time or the number of the transmitted DQ bits with 'low' level. Therefore, our BIST scheme tests output timings for various data patterns shown in Figure 4.2 to examine timing variations due to SSO noise. In order to show the effect of SSO noise on timing, DQS to DQ per-pin skew variations are measured.

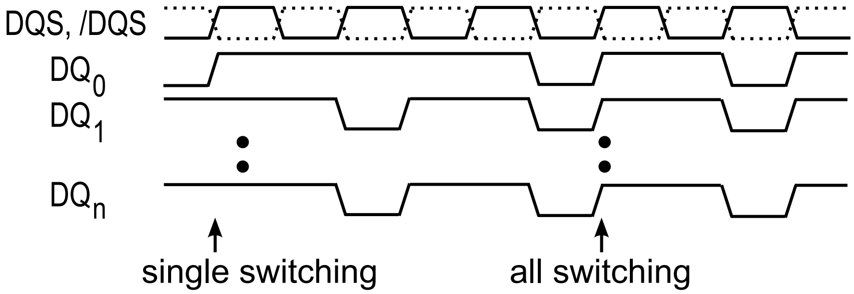


Figure 4.2: DQS and DQ data pattern generation

4.2.2 Circuit Structure for Output Timing Test

We implemented a circuit under test (CUT) unit shown in Figure 4.3 to verify our BIST scheme. This CUT includes simple modelings to test memory output timing operation. In memory devices with a large number of parallel

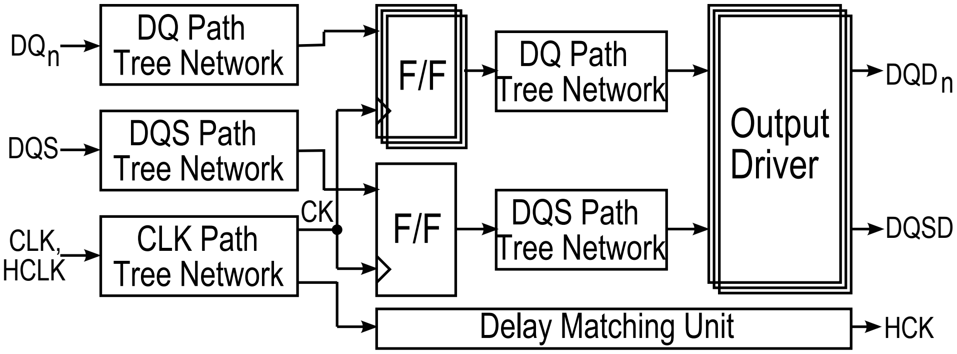


Figure 4.3: Circuit under test for output timing tests

I/Os, data skews occur due to uncertainties such as power noise, SSO and pin-to-pin mismatches, which critically affect output timings. To verify the BIST circuit, we modeled a CUT including output drivers and DQ/DQS path trees for testing data skews occurred in the CUT. Thus, the CUT includes output drivers, tree networks for DQ, DQS and CLK, and a delay matching unit for HCK. HCK is a test clock used in the BIST scheme to capture data and needs to be closely aligned with $\mathrm{DQD_n}$ and DQSD where n indicates the number of DQ pins. The delay matching unit thus simply replicates circuits used in DQ and DQS paths to match the time delay between HCK and data paths.

During a read operation, DQ and DQS are aligned by CLK for the synchronous memory system and thus skews between DQ, DQS, and CLK also affect output timing parameters. Therefore, the CUT includes a model of the CLK path as well. The aligned DQ and DQS are transferred to output drivers through tree networks. These signal paths are designed to be transferred with an identical time delay because the memory system supports parallel 16 or 32 data pins. To satisfy tight timing specifications, even small delay mismatches between DQ_n and DQS cannot be ignored. Therefore, a model of RC tree interconnection networks for clock and data paths was included in the CUT to test the delay mismatch using our BIST scheme. Moreover, to test timing variations due to slew-rate changes, the output driver in CUT was implemented as a distributed-type [14] which can adjust the transistor size using control codes. Since the CUT was implemented only to validate our BIST scheme, it has not been optimized perfectly to decrease mismatch between data and clock paths.

4.3 BIST Architecture and Design Strategies

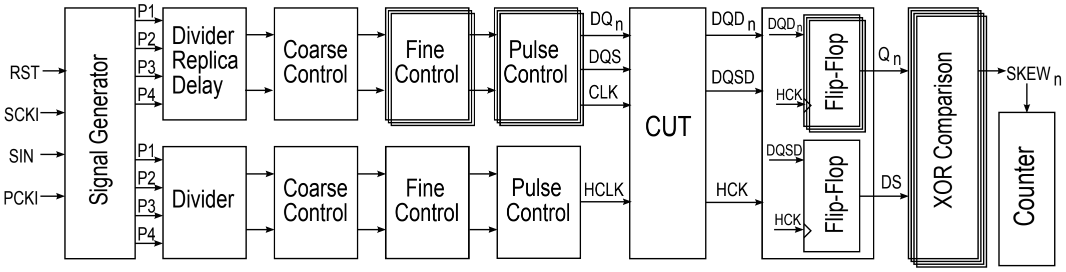


Figure 4.4: BIST architecture for testing timing skews

4.3.1 Architecture

We implemented a BIST scheme including the CUT unit shown in Figure 4.4. The BIST scheme measures the delay mismatch between DQD_n and DQSD signal paths in the CUT.

The BIST architecture consists of a ring oscillator based signal generator, divider, control units, flip-flops, comparison unit, and counters. The signal generator generates signal patterns of $\mathrm{DQ_n}$, DQS, CLK and HCLK. The coarse and fine control units control the time delay between the generated pattern and the test clock using a cycle-by-cycle method [32]. This method increases or decreases the time delay by an amount determined by the resolution of the fine control unit at every clock cycle. A phase interpolator is used in the fine control circuit to generate a small delay and a test method for finding its resolution is also presented. We use the signal generator and control circuits presented in [32]. Furthermore, we add dividers to resolve internal signal integrity. We use dividers to generate a slower test clock to control the time delay with high accuracy. However, we still use the fastest signal for the phase interpolator to generate the time delay with high resolution. Even though this method increases the test time because it uses a slower test clock and captures data once per two clock cycles, it increases test accuracy and testability for higher speed memory interfaces. Multiple dividers can be tied together to make the test clock slower and to test higher speed interface signals.

This scheme is also highly compatible with low cost ATE because all external inputs for generating test sequences are low speed signals. Figure 4.4 describes RST for chip reset, SIN for scan input, SCKI for scan clock, and PCKI for test reset. Using these input signals, we can change the coarse and fine control codes to control the time delay between HCK and $\mathrm{DQD_n}$ and DQSD. The test results for delay mismatch, $\mathrm{SKEW_n}$, are also low speed signals. Therefore, any external high speed signals are not needed in this scheme.

4.3.2 BIST Design and Test Strategies

During the test, we move the signal edge of HCK by unit delay at every clock cycle. The unit delay is determined by the coarse and fine control blocks as shown in Figure 4.5. This scheme uses 2-bit and 6-bit codes for coarse and fine control and thus the test resolution ($\mathrm{t_{PI}}$) is calculated using Equation (4.1),

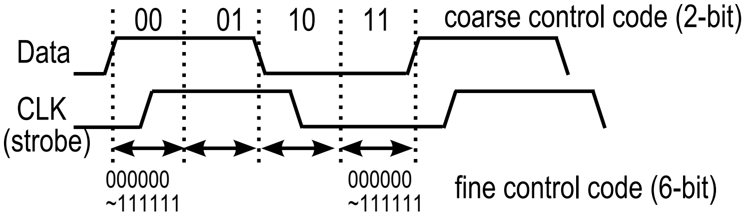


Figure 4.5: Two-level coarse and fine controls

$$\mathrm{t_{PI}} = \mathrm{t_{OSC}} \div 2^{\mathrm{N_c}+\mathrm{N_f}} \tag{4.1}$$

where $\mathrm{t_{OSC}}$ denotes the fastest clock generated from the signal generator, and $\mathrm{N_c}$ and $\mathrm{N_f}$ denote the number of bits of coarse and fine control, respectively. The resolution is tested simply using our control units and flip-flops as shown in Figure 4.6. For testing the phase interpolator, the time difference between Data and CLK is fixed with one fine control code and the phases of Data and CLK are moved in pairs by one fine control code at each test trial. Ideally, the 'high' level should be captured 63 times in one coarse control range. However, because of non-linearity of phase interpolator and jitter, it might be less than

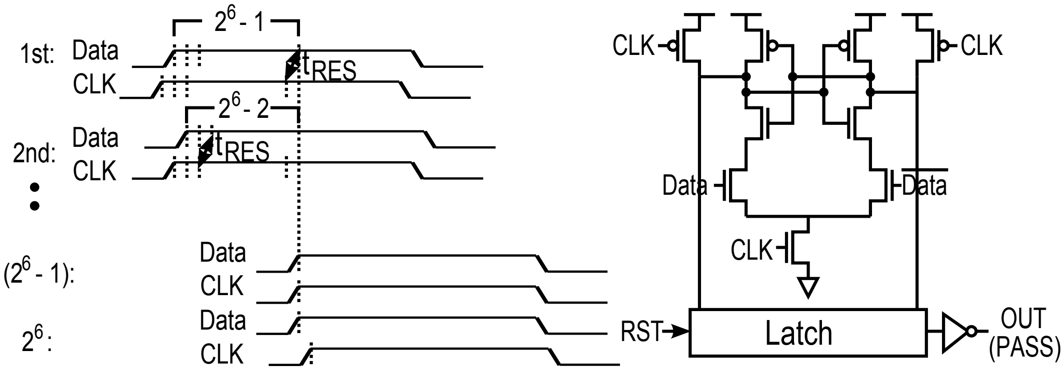


Figure 4.6: Test method for phase interpolator

63 times. By counting the number of 'test passes' in one coarse control range, we measure the average test resolution. Since the BIST scheme only needs to control the relative time delay for testing memory output timing, this test method for phase interpolator is very effective.

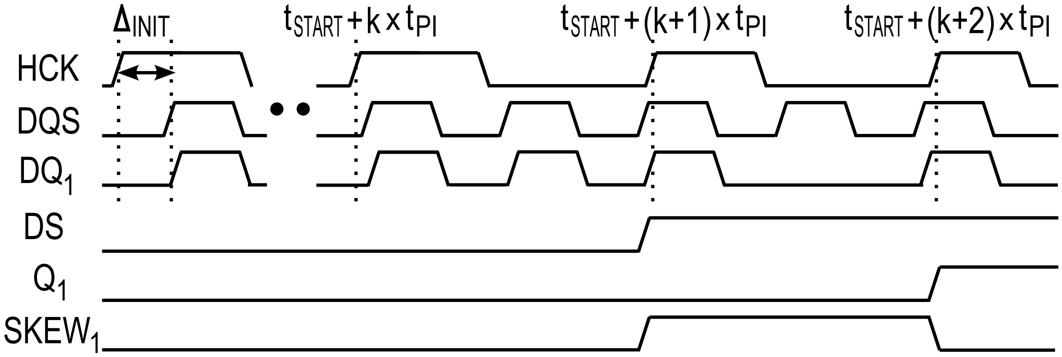


Figure 4.7: Overall logic and timing operations for testing output skews

Figure 4.7 shows the overall logic and timing operations where Δ_{INIT} is an initial time delay between data and test clock and t_{START} indicates the start time of HCK. While the time delay is controlled, data is captured by HCK. Because of delay mismatch between DQS and DQ_n, the capturing time is different causing $\text{SKEW}_1 \ldots \text{SKEW}_n$ to come out with a pulse width of tCK, where tCK indicates the test clock cycle. Therefore, the test results are analyzed by simply using counters to detect timing pass/fail and measure timing margin variations. Moreover, by comparing the SKEW_i values, this scheme can specify the data pin with the worst delay mismatch.

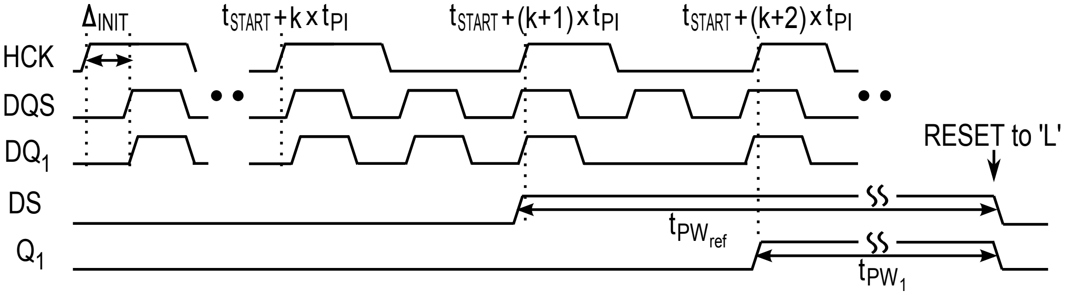


Figure 4.8: A method for testing tighter output timing parameters

Furthermore, this scheme increases testability for higher memory speed interfaces by adding another feature shown in Figure 4.8. As the speed of a memory interface increases, the test clock also needs to be faster and thus the pulse width of SKEW_n becomes narrower. Higher cost test equipment is required to measure SKEW_n especially for detecting a very small delay mismatch in high speed operation. Therefore, we measure pulse widths of DS and Q_n instead of SKEW_n. Figure 4.8 shows the timing operation. The key idea of this scheme is to reset the second transitions of DS and Q_n to reform them to a pulse. The second transition time is controlled by RST and thus this scheme can decide the minimum pulse width. We count the number of tCK cycles to measure the pulse widths ($t_{\text{PW}_{\text{ref}}}$, t_{PW_n}) where the difference of tCK cycles indicates skew. This method increases the compatibility with slower testers.

4.4 Experimental Results

4.4.1 Test Flow

Figure 4.9 shows the procedure of testing output timing using the BIST scheme. The test procedure is done in following steps: phase initialization for the signal generator, data pattern generation, setting time delay and test modes, and analysis of test results.

The test starts with phase initialization of the signal generator and data pattern generation. In the next step, initial delay, Δ_{INIT}, is set using the coarse control according to the test requirements (for example, smaller Δ_{INIT} decreases test time).Timing test can be done using fix mode or self mode: fix mode is used for testing timing pass or fail, while self mode is used to measure timing margin/variations. In fix-mode test, timing is tested with a fixed time delay; in self-mode test, delay is changed by a minimum resolution at every clock cycle. To detect timing pass/fail, we set Δ_{INIT} to a known timing specification (t_{SPEC}) and select fix-mode test. To measure timing margin/variations, we set Δ_{INIT} to '$\alpha \times$ tCK' or '$\beta \times$ tCK' and select self-mode test. The '$\alpha \times$ tCK' is used when timing has failed for t_{SPEC} and it is larger than β, which is used when timing has passed for t_{SPEC}. For our experiments, we used α equal to 0.25 and β equal to 0.1. The delay is changed by the minimum test resolution (t_{PI}) at each step (k) of the self-mode test.

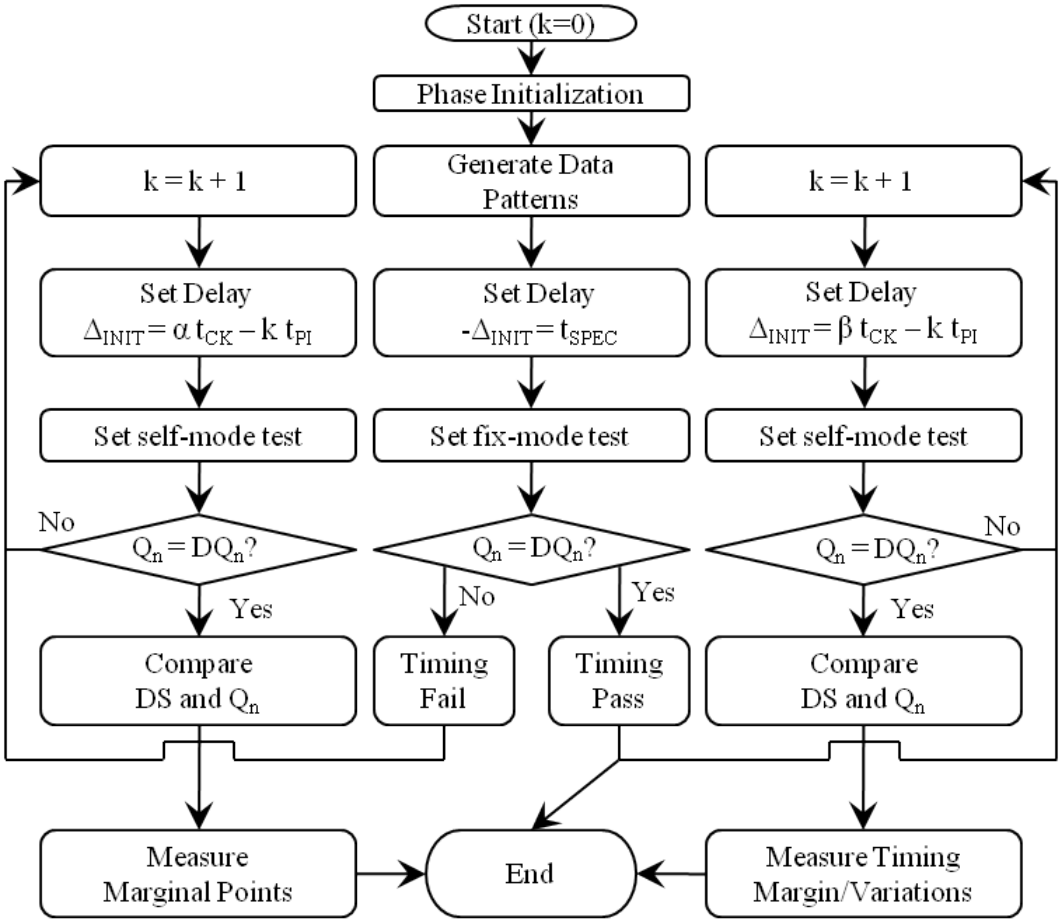
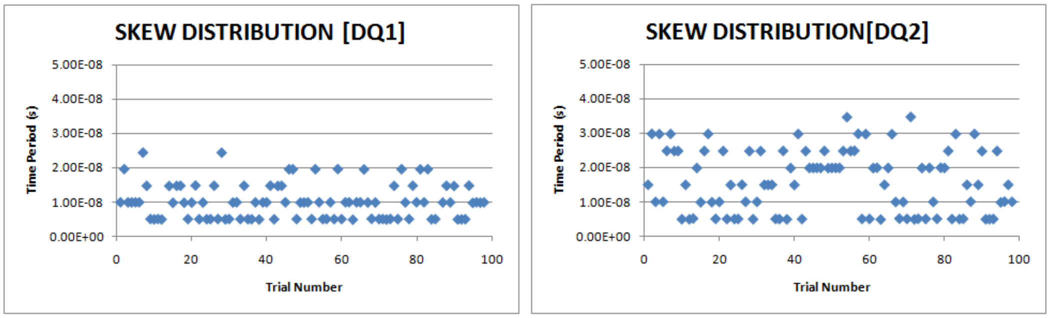


Figure 4.9: Procedure for timing test

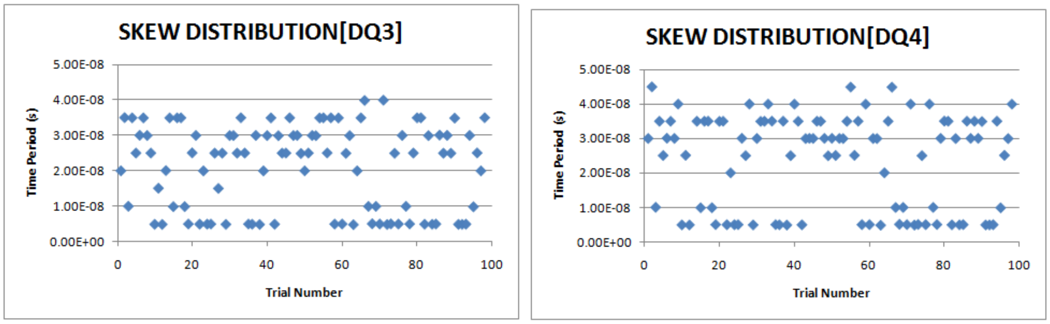
This process is repeated until data captured by the BIST scheme is the same as the expected data. This procedure gives us timing margin/variations.

4.4.2 Simulation Results

In the source synchronous DDR memory interface, output timing skews and duty distortions, the most stringent parameters, are the main hindrances for high speed performance. During the read operations, the data signals DQ and DQS are synchronized with an external clock using DLL. Therefore, all output skews are caused by the DLL jitter and duty distortions of the data signals. Since the rising edges of data signals are synchronized with the clock generated by DLL, the DLL jitter causes skews in the rising edges if there are enough margins between data signals and the DLL clock. The skews in the falling edges are caused by duty distortions of the data signals.



(a) SKEW = 5 Cycles (10ps) (b) SKEW = 7 Cycles (14ps)



(c) SKEW = 8 Cycles (16ps) (d) SKEW = 9 Cycles (18ps)

Figure 4.10: Monte-Carlo simulation results for skew detections

Figure 4.10 shows Monte Carlo simulation results (100 trial) for skew detections. We insert around 5ps skews intentionally between DQS and DQ using parasitic RC delays for simulations, and then measure the relative skews between DQ signals. We used dividers to slow down the operating speed of the testing circuits for measuring simultaneously skews at high and low edges. We assume I/O operations at 1.6Gbps data rates and so the clock frequency of testing circuits slows down to 200MHz by using dividers twice. Accordingly, the 'SKEW' pulses are formed with a unit of 200MHz as shown in the y-axis of Figure 4.10. The pulses have different values at each trial by variability simulations. The skew distributions by the monte carlo simulations are shown as 5, 7, 8, and 9 cycles where one cycle means the resolution of the phase interpolator. We assume the phase interpolators are guaranteed to have a 2ps resolution. As a result, the skew variations have the values of 10ps, 14ps, 16ps and 18ps, respectively, for four DQ signals. In this specification, we determine the performance of this system in the worst case, which is 18ps.

4.4.3 Chip Implementation and Test Setup

We extended our work for testing output timing parameters using the test chip configuration shown in Figure 4.11. For this test chip, we modified our previous BIST scheme [36] and added CUT blocks for output timing path modelings. This test chip was implemented and fabricated with a 0.18-μm CMOS process. Figure 4.12 shows the chip die photo including the CUT and BIST schemes. The total chip size is *'800 μm × 130 μm'* excluding bond pads and ESD circuitry, out of which *'300 μm × 140 μm'* is dedicated to the BIST scheme. The area is small compared to other on-chip timing

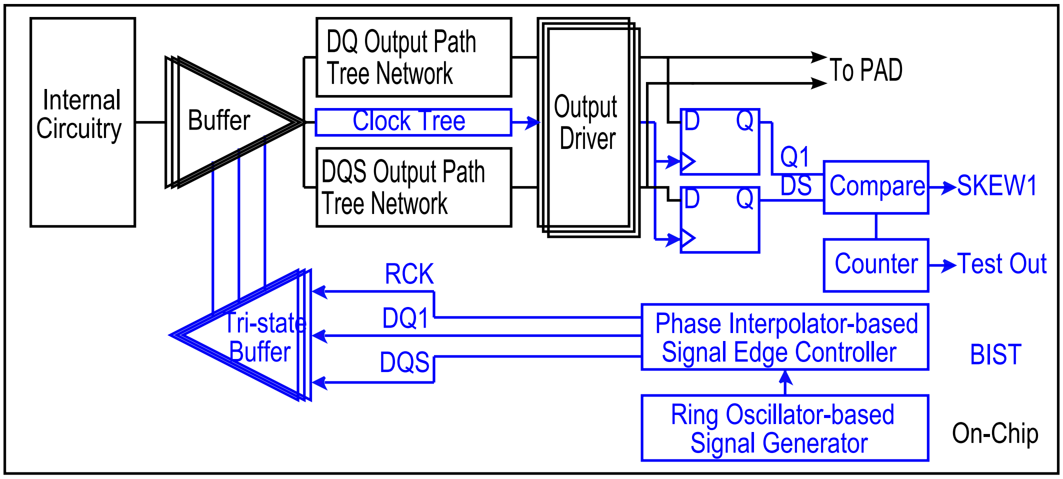


Figure 4.11: Configuration for testing data skews

test circuits [70], [13]. To test this chip, test equipment is setup as shown

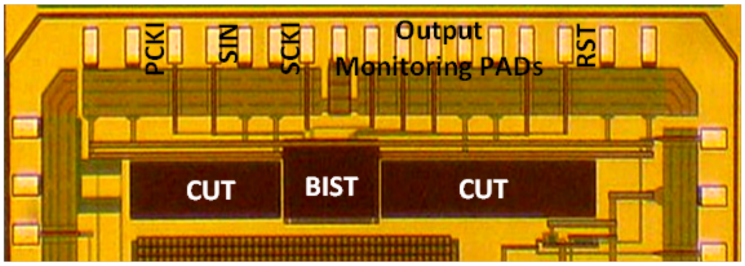


Figure 4.12: Die photo in TSMC 0.18-μm CMOS technology

in Figure 4.13. A low cost PC-based Janateck LA-Logic pattern generator (Maximum operating frequency is 10 MHz) is used for generating test input sequences. For test result measurements, output monitoring PADs are structured in test chip to probe internal clock signal and test results of timing pass/fail and timing variations. Low speed logic analyzer and digital oscilloscope are used to analyze test results and monitor waveforms.

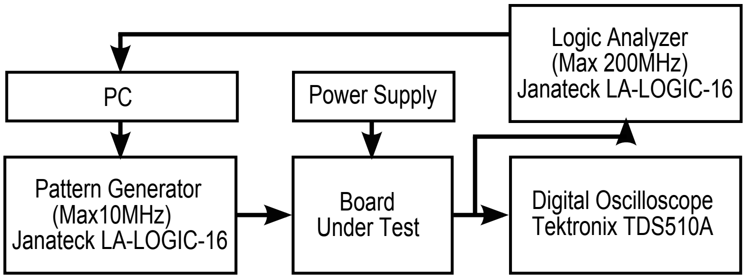


Figure 4.13: Test setup

4.4.4 Measurement Results

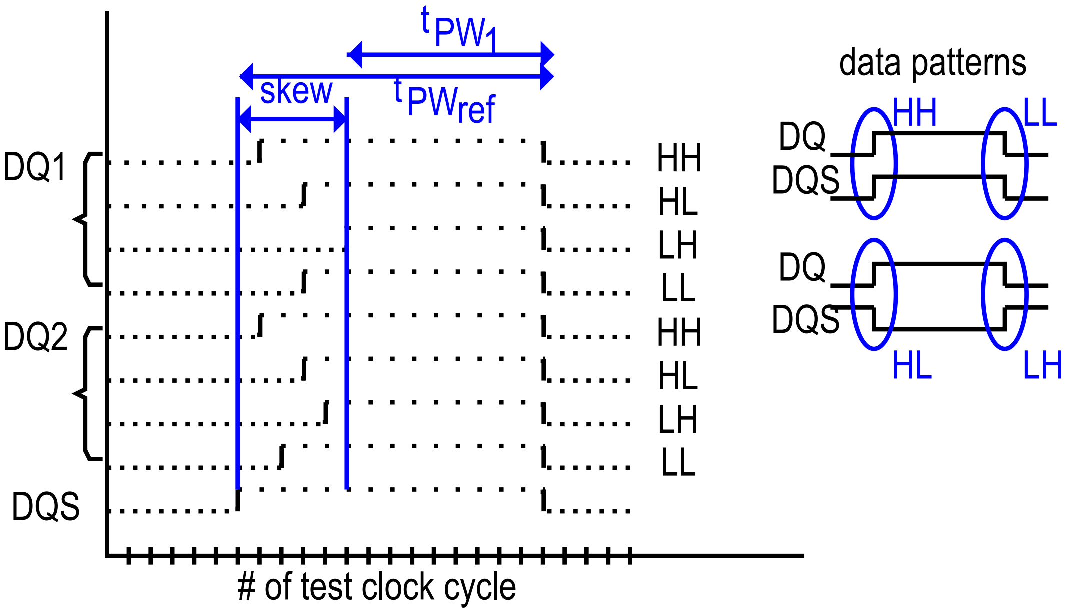


Figure 4.14: An example of test results

Figure 4.14 shows an example of the test results that measure data skews depending on the data patterns. To test data skews considering the DDR operations, the data patterns showing the different rising and falling transitions between DQ and DQS are applied. The test results show a pulse width with a multiple of the test clock cycle. The maximum difference among the pulse widths indicates the data skew occurred in the CUT. In the example, the skew of five test clock cycle occurred, which indicates a data skew of 100 ps (one test clock cycle indicates a 20 ps time delay). Based on this concept, the detailed measurement results are following.

We designed a ring oscillator with a period of 1.25 ns (t_{OSC}) and a divide-by-4 (N_{div}) circuit to generate the test clock with a period of 5 ns (tCK). Figure 4.15 shows a probing result of HCK with 1.8 V power supply. From the measurement, tCK is 5.086 ns and is a little different from the simulation result because of a noisy probe channel and ring oscillator jitter itself. However, the test clock does not affect test accuracy because we measure the relative timing variations between DQS and DQ paths by using a common test clock. On the other hand, tCK has very small effect on test resolution as shown in

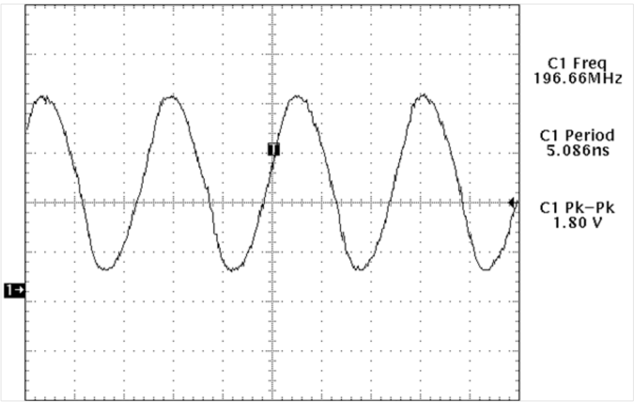


Figure 4.15: HCK clock waveform

Equation (4.2):

$$t_{\mathrm{PI}} = t_{\mathrm{OSC}} \div 2^8; \qquad t_{\mathrm{OSC}} = \mathrm{tCK} \div N_{\mathrm{div}} \tag{4.2}$$

Based on the result of tCK, the test for phase interpolator resolution is performed as shown in Figure 4.16.

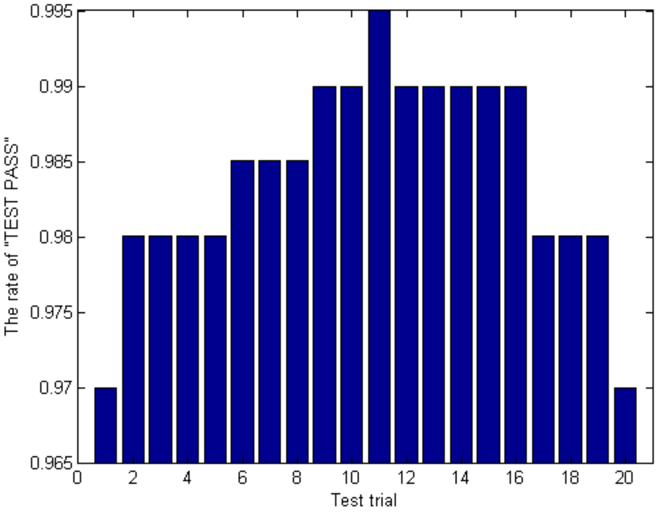


Figure 4.16: Test results of phase interpolator

We performed 20 test trials and each test trial includes 63 times possible 'TEST PASS' result. We calculate the rate of obtaining 'TEST PASS' at each test trial as shown in Figure 4.16. Thus, 10 ps is obtained as the average test resolution while the maximum error is 3 % and the average error is 1 %. Therefore, the performance of on-chip signal generator has little effects on test resolution and accuracy for this BIST scheme, and thus less design efforts are required.

Timing test is performed using fix and self modes. Table 4.1 shows a test result for one DQ in fix mode. At each test trial, we change the control code and determine timing pass/fail and then read the control code at the transition from pass to fail.

Test trial	Code	Test result
1	100000	PASS
⋮	⋮	⋮
5	100100	PASS
6	100101	FAIL

Table 4.1: Fix-mode test

Fix-mode test result shows that a 5-code difference ($\mathrm{N_{code}}$) is needed for timing pass. The code difference is used to calculate the required timing margin, $\mathrm{t_{MF}}$, as shown in Equation (4.3), and thus $\mathrm{t_{MF}}$ of 50 ps is obtained.

$$\mathrm{t_{MF}} = \mathrm{N_{code}} \times \mathrm{t_{PI}} \tag{4.3}$$

Table 4.2 shows the self-mode test results of $\mathrm{SKEW_{1,2}}$ for $\mathrm{DQ_1}$ and $\mathrm{DQ_2}$ depending on data patterns. More 'High' to 'Low' or 'Low' to 'High' data transitions degrade output timing performance because of switching noise and

n	Data Pattern	$\mathrm{SKEW_n}$
1	HHLL	1 tCK
2	HHLL	3 tCK
1	HLHL	2 tCK
2	HLHL	5 tCK

Table 4.2: Self-mode test

duty cycle distortion. Test results, $\mathrm{SKEW}_{1,2}$, are used to calculate $\mathrm{t_{MS}}$ indicating timing variations as shown in Equation (4.4):

$$\mathrm{t_{MS}} = \frac{\mathrm{MAX}[\Delta \mathrm{SKEW_n}]}{\mathrm{tCK}} \times \mathrm{t_{PI}} \tag{4.4}$$

The test results show that the maximum difference of $\mathrm{SKEW_n}$ is 4 tCK and thus $\mathrm{t_{MS}}$ of 40 ps is obtained.

The internal signals of DS and $\mathrm{Q_7}$ are also monitored as shown in Figure 4.17.

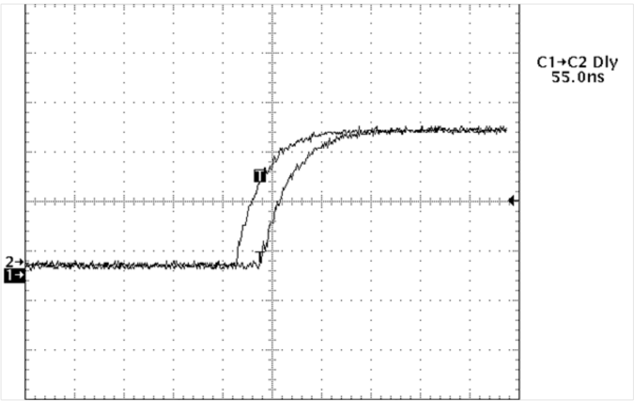
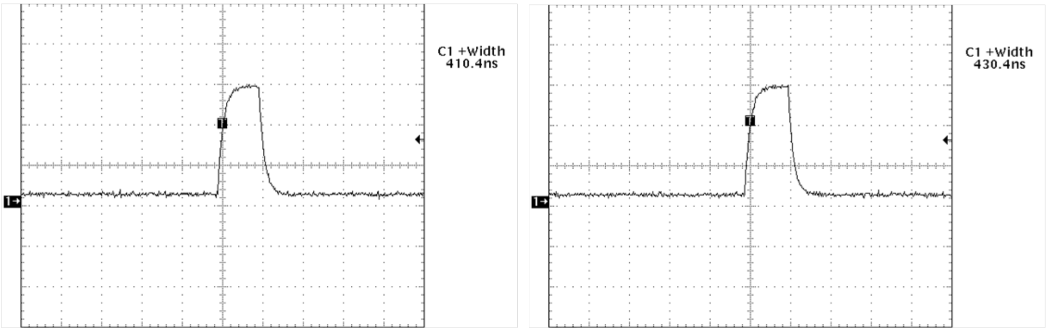


Figure 4.17: DS and $\mathrm{Q_7}$ waveform

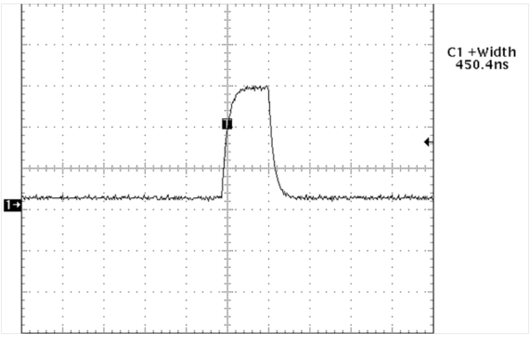
Figure 4.17 shows a timing delay of 55 ns between DS and $\mathrm{Q_7}$ which corresponds to 11 tCK and thus the time delay is calculated to be 110 ps skew.

Instead of switching noise, this scheme also is applied to measure output timing variations due to slew rate (SR) change.

$$\mathrm{SR} = \frac{\Delta \mathrm{V_{PP}}}{\Delta \mathrm{t_D}} \tag{4.5}$$



(a) HHLL-DQ$_1$ (t_{PW}=410.4 ns) (b) HLHL-DQ$_2$ (t_{PW}=430.4 ns)



(c) HLHL-DQ$_2$ + SR Change (t_{PW}=450.4 ns)

Figure 4.18: Timing variation measurement results in self-mode test

Since slew rate is related to output swing level (V_{PP}) and time delay (t_D) as shown in Equation (4.5) [14], it also affects output timing performance. For testing the effect of slew rate on output timing, we controlled the strength of the distributed-type output driver to change the slew rate. For measurements, we make DS and Q_n a pulse using the reset signal, RST. With this scheme, t_{MS} is configured as shown in Equation (4.6):

$$t_{MS} = \frac{MAX[\Delta(t_{PW_n}\ ,\ t_{PW_{ref}})]}{tCK} \times t_{PI} \tag{4.6}$$

Using this method, output timing variations due to slew-rate and data pattern changes are measured. Figure 4.18 shows test results by probing $\mathrm{Q_n}$ pulses. The pulse width is not an exact multiple of tCK because of an additional delay of the reset logic. However, it does not affect the test accuracy because we count the number of cycles and the additional delay is much smaller than one tCK. Therefore, we obtained pulse width variations of 410.4 ns, 430.4 ns and 450.4 ns for $\mathrm{DQ_1}$ and $\mathrm{DQ_2}$ by changing the slew rate and data pattern. Each pulse width corresponds to 82 $\mathrm{t_{CK}}$, 86 tCK and 90 tCK. Thus, the maximum time difference is 8 tCK and thus $\mathrm{t_{MS}}$ of 80 ps is obtained. The time variation of 40 ps occurs due to $\mathrm{DQ_1}$-to- $\mathrm{DQ_2}$ skew and different data pattern, and the time variation of 40 ps occurs due to slew-rate change. These measurement results show our BIST scheme can be effectively applied to test output timing variations due to mismatch factors.

4.5 Conclusion

This chapter presented experimental results on fabricated test chip with a built in self test and diagnosis scheme that measures memory output timing variations using a low cost tester. In this technique data patterns are generated using a ring oscillator based on-chip signal generator. The time delay between data and clock is controlled using a cycle-by-cycle control method. We also implemented a circuit under test by replicating output paths used in memory systems to validate the BIST scheme. Moreover, this technique does not need any high speed signals for testing high speed output timing. The test stimulus is generated using slow signals of 10 MHz and the outputs of this chip are a multiple of the test clock period, and hence slow. These characteristics allow us to test high speed memory output timing with high accuracy and resolution using a low speed and low cost tester. Using 0.18-μm CMOS technology, this scheme was fabricated and we measured the test resolution of 10 ps. With this resolution, measurement results for output timing variations due to slew-rate change, pin-to-pin skews and data pattern are presented in this chapter.

Chapter 5

On-Chip Delay Line Based Timing Sampler for DDR Timing Tests

5.1 Introduction

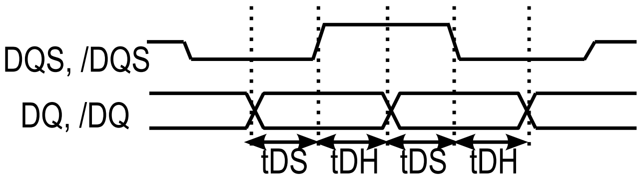
As the memory speed has increased, the internal circuitry has become more susceptible to timing failures. Accordingly, accurate and precise test techniques are required to detect small timing errors for testing the I/O timing parameters. In Chapter 2 and Chapter 4, the on-chip timing sampler produces the data and strobe signals, and then controls the time difference between the signal edges of the data and strobe. The time difference is calculated to measure the input setup and hold time, and output data skews. The strobe-scanning method using the phase interpolator needs the complicated control method, and thus it takes longer test times for memory devices operating in a wider frequency range. In this chapter, we present another on-chip timing sampler to improve testability and controllability with lower hardware overhead. This chapter thus presents a delay line based on-chip timing sampler to measure a time delay without any limitations of the external clock frequency. This scheme was designed using fully digital circuits and supports a digital calibration to control the timing edges for sampling. This approach also does not require the additional complicated control circuits to generate the read and write interface timings.

For the delay line based on-chip timing sampler, we developed a programmable double-capture generator (PDCG) by modifying the programmable capture generator (PCG) presented in [62]. Unlike the PCG, the PDCG generates the dual-capture signal that has controllability for both rising and falling transitions simultaneously. In the DDR operations, the rising and falling edge transitions of data and clock operate in pairs and thus affect the timing parameters at a time. Therefore, our work effectively tests the DDR I/O timings because the dual-capture approach is close to the real interface timing operations. Moreover, the dual-capture enables the testability for the timing degradation due to the duty distortions. Additionally, the technique produces a fast test clock regardless of the external clock frequency, and thus does not need a high-end ATE. We also present a scheme to provide a dynamic delay range by modifying the PDCG. This scheme is to give controllability for the resolution since we do not need to test with high resolution for low-end memory devices. On the other hand, we replace the on-chip timing sampler presented in Chapter 2 with this delay measurement method and then combine with the low-cost measurement method presented in Chapter 3. The scheme is thus valuable, not only for small delay detections but also for large delays. Moreover, our work can be incorporated into the current scan based delay test methods to measure the time delay, and thus is available for timing characterizations during post-silicon validation. Therefore, this scheme validates the variability of the circuit timing operations caused by process parameters, temperature, cross-talk coupling, and supply voltage noise.

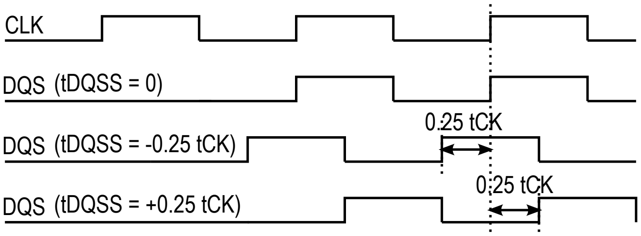
5.2 Design Background

For DDR memory devices, the most stringent interface timing parameters are defined among DQS, DQ and CLK signals. The timing specifications are illustrated in Figure 5.1. In order to sample data properly into registers, the DQ needs to arrive the I/O pin before the DQS arrives, and maintain valid after the DQS is asserted. Figure 5.1(a) shows the timing relationship between DQ and DQS. The setup time (tDS) indicates the time delay between the arrival time of DQ and the rising/falling edges of DQS assertion. The hold time (tDH) is defined as the time delay between the DQS assertion and the valid DQ period after DQS is asserted. Another input timing parameter, tDQSS, states the timing relationship between DQS and CLK. The DDR memory devices have two clock domains of DQS and CLK, and the DQS domain should be changed to the CLK domain internally. For this requirement, DQS must maintain a certain timing relationship with CLK. The time delay between the rising edges of CLK and DQS is defined as tDQSS as shown in Figure 5.1(b). The most tough output timing parameter, tDQSQ, specifies the allowed skew range between DQS and its associated DQ signals. All these timing parameters need to be guaranteed both at the rising and falling edges of DQS (DDR function)

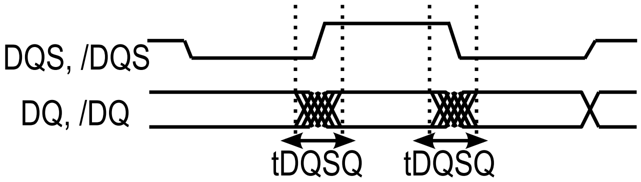
Table 5.1 shows an example of DDR3 memory interface timing specifications [2]. It explains that timing parameters become tighter as the operating



(a) Setup and hold times (tDS/tDH)



(b) Clock and data-strobe skews (tDQSS)



(c) Data-to-data/data-to-clock skews (tDQSQ)

Figure 5.1: Memory interface timing parameters

speed increases where tCK indicates the external clock (CLK) frequency. For testing these timing parameters, a key challenge is to generate a proper timing delay among data, data-strobe and clock. Also, a high timing resolution and accuracy need to be guaranteed. Moreover, an on-chip test circuit with a wide delay range is required to have a higher test coverage. For example, tDQSS is a clock-based timing parameter unlike the tDS, tDH and tDQSQ as shown Table 5.1. Generally, memory devices integrated in SoCs operate in a wide frequency range such as from 200 MHz to 1 Gbps. Thus, for testing all the interface timings, the testing range of on-chip test circuits needs to be changed

	DDR3-800	DDR3-1066	DDR3-1333	DDR3-1600
tDS	75 ps	25 ps	30 ps	10 ps
tDH	150 ps	100 ps	65 ps	45 ps
tDQSQ	200 ps	150 ps	125 ps	100 ps
tDQSS	±0.25 tCK	±0.25 tCK	±0.25 tCK	±0.27 tCK

Table 5.1: DDR3 memory interface timing specifications

dynamically according to the tCK value. Therefore, the key requirements of on-chip measurement circuits for testing high-speed DDR memory interfaces are high resolution, wide delay range, and low-cost.

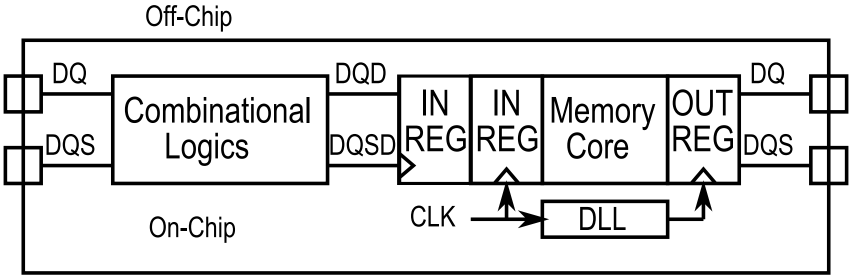


Figure 5.2: Simplified memory interface structure

Figure 5.2 shows a simplified memory interface block diagram, the interface timing parameters of tDS/tDH, tDQSQ and tDQSS are defined between DQS and DQ or DQS and CLK in the memory system. The delay between two paths needs to match perfectly to have a maximum data valid window since the delay mismatch directly degrades the timing parameters. The amount of the delay mismatch indicates the required minimum timing margin to capture the valid data. Thus, measuring the delay difference is an effective way to test the I/O timings. If the measured delay is smaller than the given timing specification, it means that the memory system meets the timing specifications.

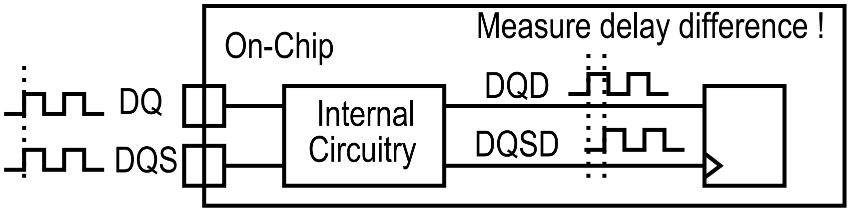


Figure 5.3: Basic idea of the delay line based timing sampler

Thus, the basic idea of our work is to measure a delay difference between DQ and DQS. Figure 5.3 is a simplified diagram explaining the basic idea. While DQ and DQS are passing through from the I/O pins to the internal circuitry, a delay mismatch between DQ and DQS occurs due to uncertainties such as a path tree mismatch, crosstalk, and power noise etc. The delay mismatch directly affects the timing parameters, and thus we measure the delay difference between DQ and DQS at the inputs of I/O registers. Moreover, for effective DDR timing test, the delay measurement is performed for both the rising and falling transitions from DQ to DQD or DQS to DQSD. The strengthes of using the delay measurement method are that there is no requirement for a phase-alignment circuit to generate any complex test timings and test vectors.

Another issue for testing memory I/O timings is to implement a wide delay range to test tDQSS parameter. As shown in Table 5.1, tDQSS is specified based on the operating clock frequency (tCK) such as '±0.25 tCK'. Thus, we need to implement larger delays to test the parameter. In general, most computer systems want the adjacent memory devices to operate in a wide frequency range. Thus, the on-chip test circuits need to implement the wider delay ranges at lower frequencies because it is a frequency-based parameter. Besides, the combinational logic delays between DQS and DQSD can be over '1.0 tCK' because of the long placement and routing of I/O circuitry when the chip size is large. Accordingly, we need to consider the wider control range for the delay measurement to increase the test coverage. Thus, we propose the modified scheme of the PDCG by combining the fixed delay and the variable delay. However, in general, as the dynamic range is wider, the area overhead is higher. Therefore, the modified PDCG needs to have a trade-off between low hardware overhead and wide dynamic range.

5.3 On-Chip Programmable Dual-Capture Timing Generator

5.3.1 Architecture

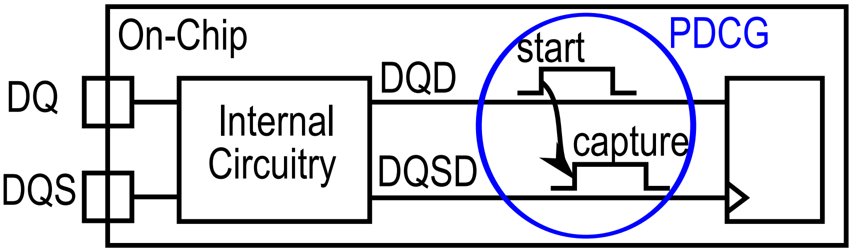


Figure 5.4: Delay measurement using PDCG

Figure 5.4 shows our overall architecture to test interface timings using the delay line based timing sampler. Using PDCG, the capture signal is generated with a certain delay with respect to the start signal. We then measure the time delay between DQD and DQSD where DQD and DQSD are the propagated signals from the I/O pins. For the circuit, we utilized the on-chip programmable capture presented in [62] and revised to apply for memory I/O timing tests. In the next section, we describe the detail structures and functions of PDCG.

5.3.2 Programmable Dual-Capture Generator

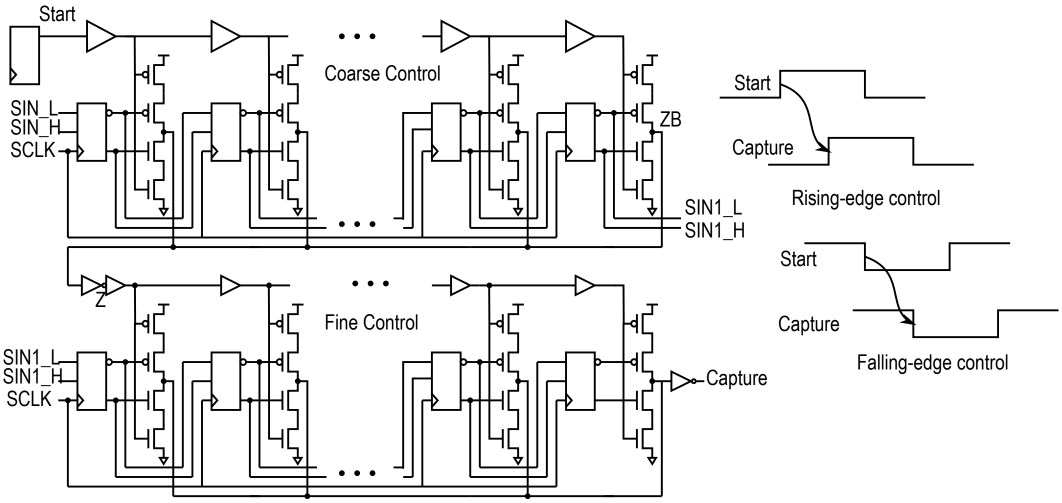


Figure 5.5: Programmable dual-capture generator

We present a scheme to test the interface timing parameters using the delay measurement method. The technique measures the delay from a launch signal to a capture signal. The capture signal is precisely controlled to measure the delay with a high resolution. The resolution can be arbitrarily changed and is not limited by the external clock frequency. Accordingly, we can test the high speed interface timing with a low-cost tester. The technique also measures the propagation delays for both the rising and falling transitions of the launch and capture signals. We read out the delay value by incorporating the generated capture signal in the ac-scan test method as shown in the previous work of [62]. Therefore, we only focus on the on-chip signal generation method. The schematic of the PDCG is shown in Figure 5.5. The 'Start' signal initiates the circuit operation and 'Capture' is the delayed signal with respect to 'Start'. After the delay control procedure is completed, the rising or the falling transition on 'Start' signal is propagated to 'Capture' signal with the programmable delay. The PDCG is applied to test the interface timing parameters in the memory system. Thus, we measure the delay difference between DQ and DQS paths. DQ or DQS is to be 'Start' signal and DQD or DQSD is to be 'Capture' signal during the delay measurement. The PDCG consists of the coarse and the fine blocks. Each block includes the delay buffers and the control circuitry using the scannable flip-flops. The two-level control method using the coarse and the fine delay buffers increases the dynamic delay range and the test resolution. Each output node of the coarse delay buffers and the scannable flip-flops is connected to both NMOS and PMOS transistors. The novel dual-capture is operated by this structure. The 'Z' signal coming from the coarse block is transferred to the fine block. The fine block precisely adjusts the total delay using a smaller delay unit. The total delay between 'Start' and 'Capture' is programmed using a series of scannable flip-flops that generate the one-hot code. The sequence order of the one-hot code determines the amount of delay. The 'SIN_L' and 'SIN_H' signals independently control the rising and falling delays. In the DDR operations, duty distortions likely occur due to the mismatches caused by capacitive loadings and cross-talk. Using the dual-capture method, we effectively detect the delay difference which occurs between DQ and DQS paths due to the duty distortions.

5.3.3 Modified Programmable Dual-Capture Generator

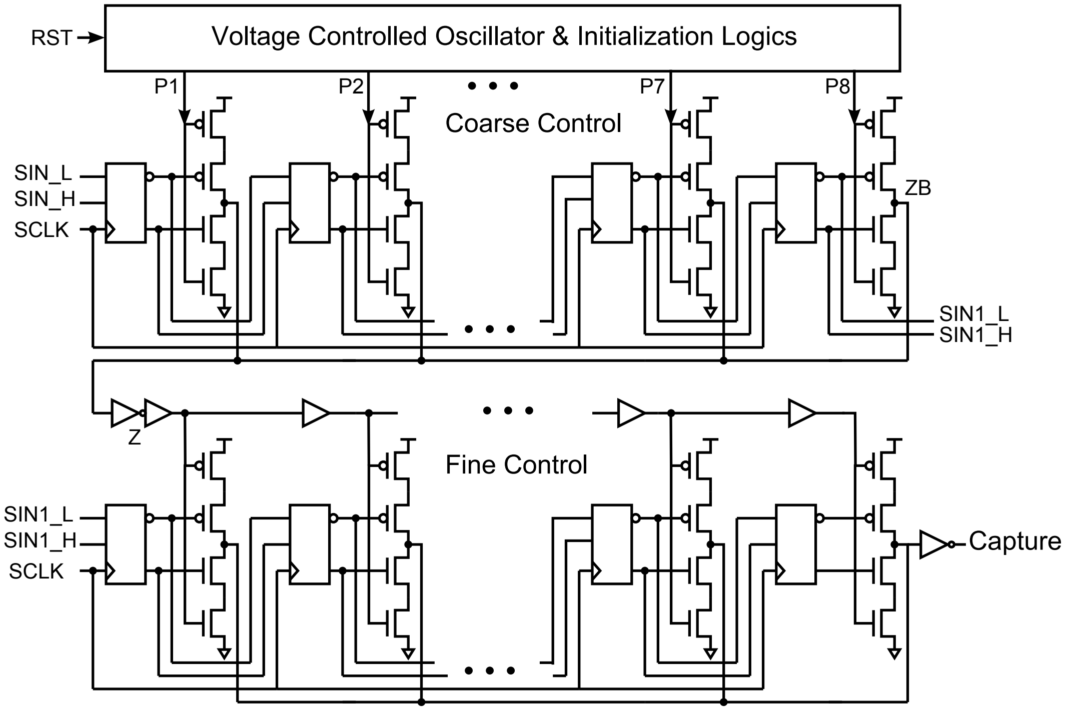
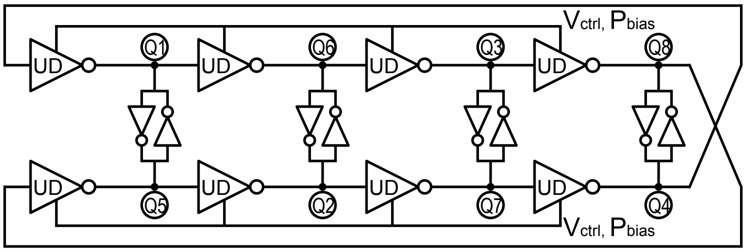
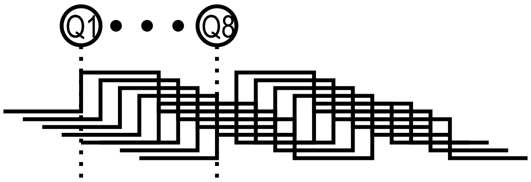


Figure 5.6: Modified PDCG

The PDCG has a reasonable dynamic delay range by combining the coarse delay and the fine delay buffers. However, the dynamic range cannot be large enough in some cases. Most computer systems want the memory devices to operate in a wide frequency range. For instance, for the DDR2 memory, the operating frequency range of 200 MHz to 533 MHz is specified [1]. In such a case, we need the on-chip signal generator with the wide operating range for testing the system with a high test coverage. Especially, tDQSS, one of the interface timing parameters in the memory system, is defined based on tCK. Thus, the dynamic range of the PCDG cannot be good enough to test the frequency-based parameter. Therefore, we propose the modified PDCG (MPDCG) to increase the dynamic delay range. The schematic is presented in Figure 5.6. The MPDCG uses the VCO instead of the coarse delay buffers.



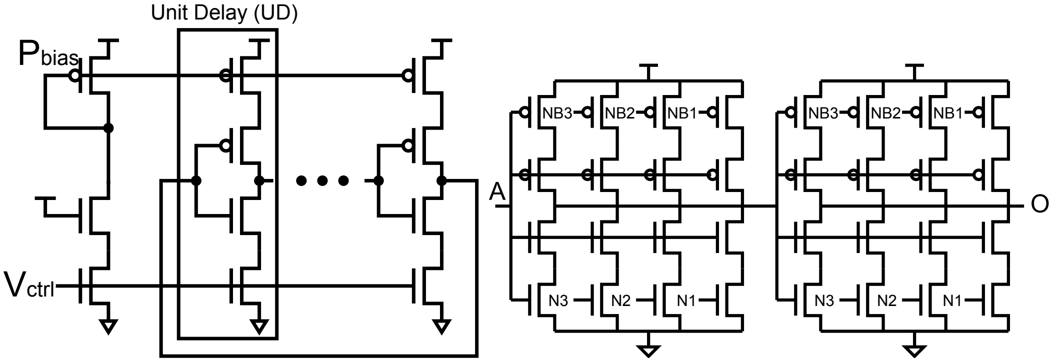
(a) Voltage controlled oscillator



(b) Octal phase generation

Figure 5.7: Signal generator

Using the VCO, we implement the variable coarse delay block with a small number of inverters. The number of the coarse delay units is the same as the PDCG for the low hardware overhead. The VCO operating frequency, tCK_{vco}, determines the coarse delay range of the MPDCG. The total dynamic range is decided by combining the delay ranges of the coarse and the fine blocks. The tCK_{vco} is controlled by the VCO control voltage regardless of the system clock frequency, tCK. Thus, the technique is not limited by the tester. The schematic of VCO is shown in Figure 5.7(a). The VCO is a pseudo differential type with the feedback inverters [7]. It produces octal phases of four differential pairs as shown in Figure 5.7(b). The unit delay (UD) cell is implemented using the current-starved type shown in Figure 5.8(a) [14]. The unit delay cell consists of PMOS and NMOS current sources, and the delay is controlled by P_{bias} and V_{ctrl}. P_{bias} is produced by V_{ctrl}. Thus, the VCO operating frequency, $\mathrm{tCK_{vco}}$, is managed by the control voltage, V_{ctrl}. Since the VCO produces octal phases, each coarse delay is '0.125 $\mathrm{tCK_{vco}}$'. Therefore, the coarse block implements the delays in the wide range with the coarse resolution. The fine delay unit is implemented using a segmented-type driver to present a smaller delay and a higher resolution. The fine delay unit is controlled by the code values of *N1*, *N2*, and *N3*. The resolution of the delay measurement is determined by the unit delay of the fine delay block. The number of delay cells selected by the scannable control signal determines the total delay value. As a result, the



(a) Coarse delay unit (b) Fine delay unit

Figure 5.8: Coarse and fine delay units

MPDCG has the low area overhead comparable to the PDCG but has a wider dynamic delay range.

5.3.4 Timing Operations

In this section we explain the overall timing operations including the initialization procedure. Figure 5.6 shows that the initialization logic is implemented to make the control easier and prevent a malfunction. Since the initial values at each output node of VCO are random, any phase of 'Q1' to 'Q8' can be produced first. Thus, the sequence order of the generated octal phases can be variable. The phase variability can initially cause a random coarse delay and thus give rise to an abrupt delay change. To solve this issue, before the delay measurement, we need an additional calibration process to correlate the sequence orders between the phase and the scannable control signals. By using the on-chip initialization logic, we prevent any extra test time. Additionally, the initialization procedure is required to set the operating period of 'Start' and 'Capture' signals to prevent conflicts between the control signals propagating through the scannable flip-flops and the phase signals generating by the VCO. The schematic of the initialization logic is presented in Figure 5.9. The basic idea is that the phases of 'Q1' to 'Q8' should be sequentially produced. Accordingly, the 'Q1' should be generated first and the reset signals are formed by 'Q1' to reset other phases. The 'RST' is asserted to initiate the VCO after the scannable control signals are propagated. The 'RQ1' is the latched 'Q1' by 'RST' and is used to generate the reset signals for 'Q2' to 'Q8'. Each reset

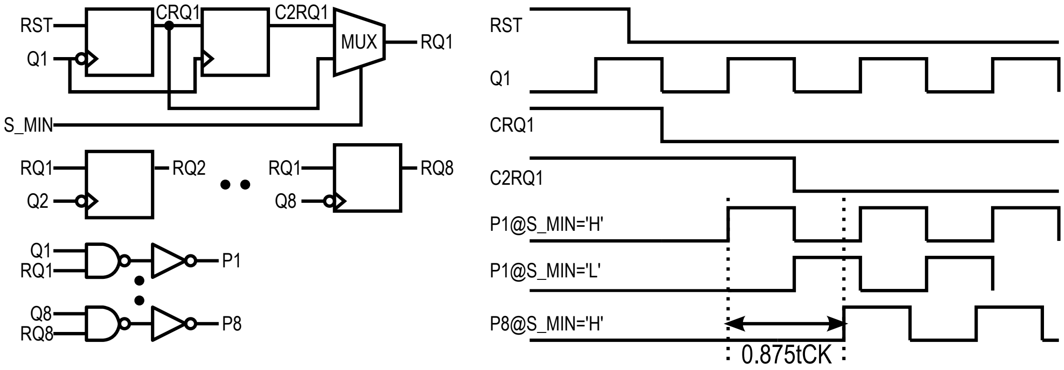


Figure 5.9: Initialization circuits

signal is separately combined with the each phase by the AND operation. Accordingly, the newly generated phases of 'P1' to 'P8' have a zero initial state. As can be seen in Figure 5.9, if the scannable control signal chooses the largest delay, the 'Capture' signal from 'P8' is delayed as much as the coarse delay of '0.875 tCK_{vco}'. In the case of 'S_MIN=L', the scheme enables the additional delay of '0.5 tCK_{vco}', which is the feature to increase the delay range.

Figure 5.10 shows the overall timing operations. The signals 'SCLK' and 'SIN' are the inputs of the scannable flip-flops. Depending on the sequence patterns of 'SIN', the numbers of the coarse and the fine delay stages are programmed. For instance, as shown in the timing diagram of Figure 5.10, the fifth coarse delay and the second fine delay are selected by the scannable control signal. Thus, the 'Capture' signal is delayed as much as the addition of five coarse delay buffers and the two fine delay buffers. Since we set the same controls for the rising and the falling transitions, the 'Capture' signal has a 50 % duty ratio. If the delay controls are differently set, the 'Capture'

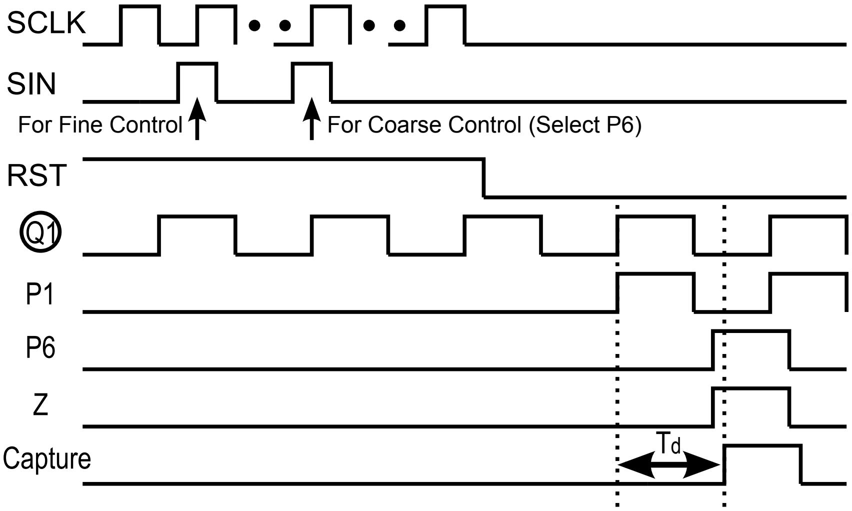


Figure 5.10: Timing operations

is produced with a non-symmetric duty ratio. Therefore, we can generate a signal pattern with controllable duty ratio.

5.3.5 Post-silicon Timing Validation

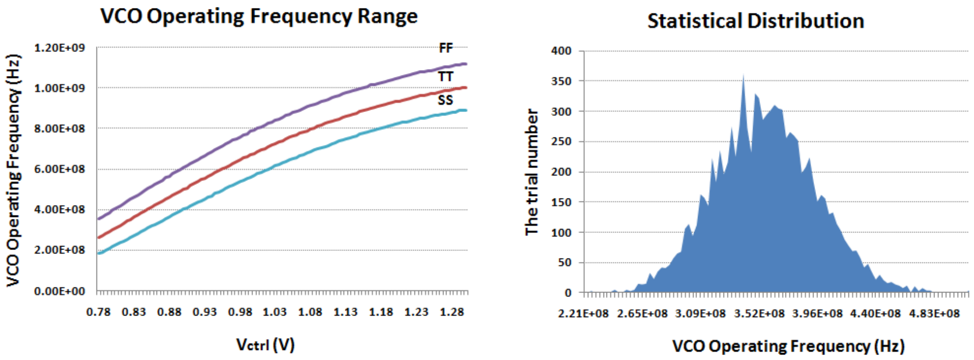
The interface timings are affected by various uncertain factors such as power supply noise, crosstalk, impedance mismatch, and the capacitive loadings. The uncertainties cause delay variations, data skews, and duty cycle distortion, which affect critically the timing parameters. Thus, the timings need to be validated under all possible conditions. However, modeling the relationship between the uncertain factors and timing failures is intractable because of their complex interactions and the uncertain root causes of the failures. On the other hand, the relative delay variation between data and clock paths is more important than the delay variation at each path in the source-synchronous interface. Therefore, the relative delay measurement between data and clock paths can be a good solution instead of using complicated test vectors. The measurement can be performed while the 'Capture' signal is controlled step by step until the timing specifications are met. The measured delay is compared with the estimated value and thus we validate the timing relationship between data and clock. In [52], a method to measure the path delay using scan chains was proposed for the design validation. However, the scheme needs to be inserted between the combinational logic and thus causes extra routing overhead. Moreover, it cannot detect the timing failures due to variations on the clock skew. On the contrary, the proposed PDCG technique works with lower hardware overhead. It also detects the timing failures due to data and clock skews. Moreover, the technique measures the timing margin by comparing the measured delay with the reference value given by the specifications. Additionally, we validate the timing failures between two paths due to duty cycle distortions by sweeping the 'Capture' signal with different controls for the rising and falling transitions.

5.4 Simulation Results

We have implemented the PDCG and MPDCG using the 0.18 μm CMOS process. In the PDCG, the delay between 'Start' and 'Capture' is measured. The total propagation delay, $T_{d,PDCG}$, is determined by combining the coarse delay, t_{ci} , and the fine delay, $t_{fj,\ k}$. Therefore, $T_{d,PDCG}$, is obtained by Equation (5.1),

$$\begin{gathered} T_{d,\mathrm{PDCG}} = t_{\mathrm{ci}} + t_{\mathrm{fj,\ k}} + t_{\mathrm{offset}} \\ t_{\mathrm{ci}} = i \times t_{\mathrm{cp}};\ t_{\mathrm{fj,\ k}} = j \times k \times t_{\mathrm{fp}}; \\ \mathrm{i} = [1,\ N_c],\ \mathrm{j} = [1,\ N_{\mathrm{f1}}],\ \mathrm{k} = [1,\ N_{\mathrm{f2}}] \end{gathered} \tag{5.1}$$

where t_{offset} is the constant offset delay due to the capacitance loadings. t_{cp} is the one coarse unit delay and t_{fp} is the one fine unit delay of the segmented-type driver. N_c and N_{f1} are the numbers of the coarse and the fine delay units, respectively. N_{f2} is the segmented number per one fine delay unit. i, j, and k are the programmed numbers of the coarse delay block, the fine delay block, and the segmented-type driver, respectively. For the MPDCG, $T_{d,MPDCG}$ is



(a) VCO operating range

(b) Monte-Carlo simulation

Figure 5.11: Signal generator simulation results

similarly represented by Equation (5.2).

$$T_{d,\mathrm{MPDCG}} = t_{\mathrm{ci}} + t_{\mathrm{fj,\ k}} + t_{\mathrm{offset}} \tag{5.2}$$

$$t_{\mathrm{ci}} = (i-1) \times t_{\mathrm{cm}};\ t_{\mathrm{fj,\ k}} = j \times k \times t_{\mathrm{fm}};$$

$$\mathrm{i} = [1,\ N_c],\ \mathrm{j} = [1,\ N_{\mathrm{f1}}],\ \mathrm{k} = [1,\ N_{\mathrm{f2}}]$$

$$t_{\mathrm{cm}}=\frac{1}{f_{\mathrm{vco}}}=\frac{I_{\mathrm{D}}}{N_{\mathrm{vco}}C_{\mathrm{tot}}V_{\mathrm{ctrl}}}$$

The values N_{vco} and f_{vco} indicate the number of VCO delay stages and the VCO operating frequency, respectively. I_D is the current flowing in the coarse UD and C_{tot} is the output capacitance at each node. t_{cm} is the variable VCO unit delay that enables the wide dynamic delay range. t_{fm} is set to the same value as t_{fp}. For both PDCG and MPDCG, N_c, N_{f1}, and N_{f2} are selected to 8, 3, and 3 stages, respectively. Since the MPDCG is the extended scheme of PDCG, the simulation results of MPDCG are shown in this section. For the coarse delay, the VCO simulation results are shown in Figure 5.11. As can be seen, the VCO operating range is from 250 MHz to 1 GHz. The range is determined by the timing specifications of the system that we want to test. Thus, the minimum value of the coarse delay unit is '0.125 $\mathrm{tCK_{vco}}$', which is 125 ps. Figure 5.12 shows the simulated waveform of the coarse delay block. The output of the coarse block, 'Z', is delayed as much as '0.25 $\mathrm{tCK_{vco}}$', 250 ps, by selecting the 'P3' phase. Therefore, 125 ps is the coarse test resolution. However, in reality, the actual resolution is highly dependent on the process variations. The delay variations of data and clock paths can be changed with the same or the opposite directions. Therefore, Monte Carlo simulations are

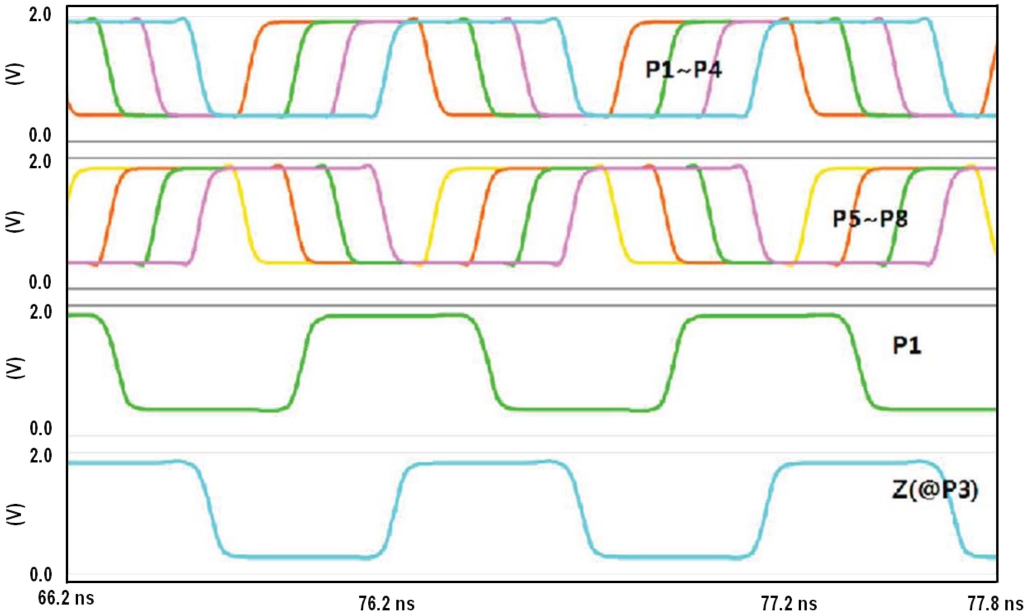
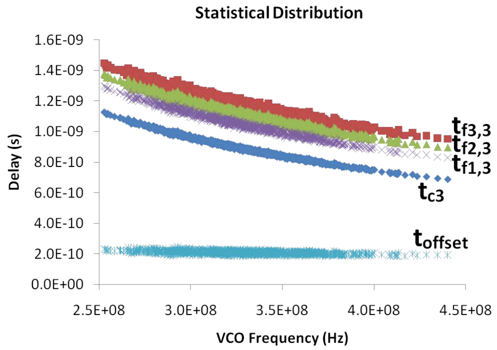
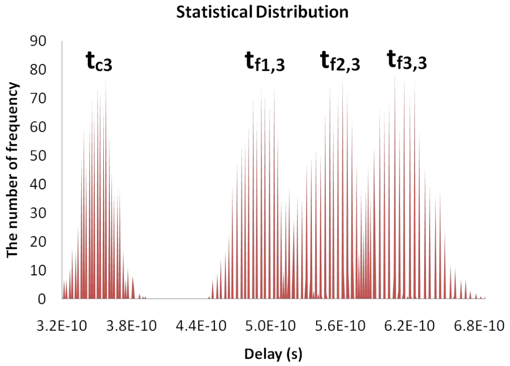


Figure 5.12: Simulated waveform

performed to consider the process variations during the design. Based on the variability simulations, the number of the delay stages and the buffer sizes can be optimally selected. Figure 5.13 shows the Monte Carlo simulation results for the delay variations. The simulation results present the statistical distribution of the total delay depending on the programmable coarse and the fine delay controls. For the coarse delay block, we show the delay changes as V_{ctrl} is changed. Figure 5.13(a) shows that the delays are linearly changed depending on the VCO frequency. As presented in Equation (5.2), t_{c3} indicates the third stage of the coarse delay block, and $t_{f1,3}$, $t_{f2,3}$, and $t_{f3,3}$ indicate the third segmented-type driver at each fine delay unit. The total delays are shown according to the each programmable coarse and fine controls. When t_{c3} is



(a) Frequency vs. Total delay



(b) Total delay (@tCK$_{\text{vco}}$ = 1 ns)

Figure 5.13: Delay variation simulation

applied, where tCK$_{\text{vco}}$ is 2.5 ns, the total delay is around 750 ps by adding the average coarse delay of 625 ps and the coarse offset delay of 125 ps. When $t_{f1,3}$ is applied, the total delay of around 890 ps is obtained by adding the coarse delay of 750 ps, the fine delay of 60 ps, and the fine offset delay of 80ps. $t_{f1,3}$, $t_{f2,3}$, and $t_{f3,3}$ are the results of adding 3, 6, and 9 fine delay units, respectively, where the fine delay unit is 20 ps. Thus, the test resolution is 20 ps but it can be higher with the scaling of the technology node. Figure 5.13(b) shows the total delay distribution where $\mathrm{tCK_{vco}}$ is 1 ns. Because of the offset delay of 205 ps (t_{offset}), the minimum delay is limited to 't_{offset} + 20 ps'. Therefore, the dynamic delay range (DR) is determined by the minimum and the maximum values of $T_{d,MPDCG}$, and thus we achieve the DR as the following Equation (5.3).

$$\begin{aligned}\mathrm{DR} &= [\mathrm{DR_{min}},\ \mathrm{DR_{max}}]\\ &= [\mathrm{t_{offset}} + 20\ \mathrm{ps}\ ,\ 0.875\ \mathrm{tCK}_{vco} + \mathrm{t_{offset}} + 180\ \mathrm{ps}]\end{aligned} \tag{5.3}$$

In order to have the wider delay range, we can perform offset cancelation by inserting the offset delay in the launch signal path. Then, by measuring the delay from the delayed 'Start' to 'Capture' signals, we can cancel the offset delay. However, this increases the area overhead. Figure 5.14 shows the 0.18-μm CMOS layout (98 μm × 70 μm) of the PDCG to estimate overall

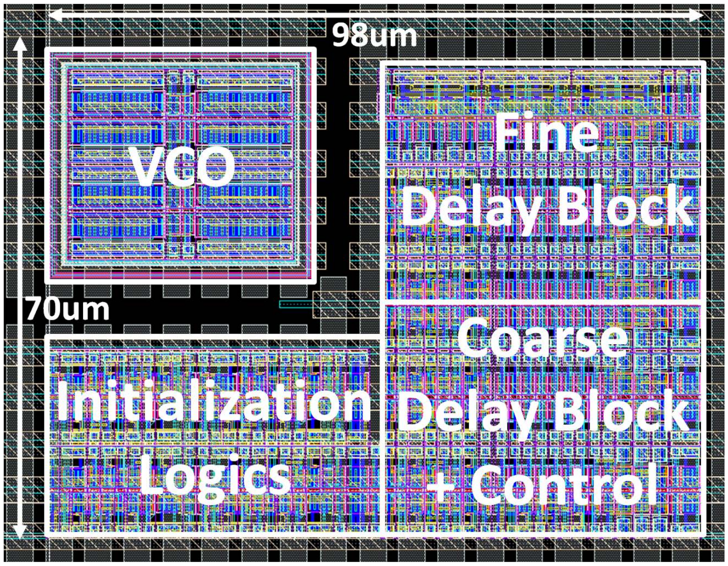


Figure 5.14: Layout

hardware overhead. If we apply the proposed PDCG to test the interface timing parameters in the memory device, the area overhead of the test circuit is under 1% since the VCO and the initialization logic can be shared for testing all DQ timings. Therefore, we obtain the advantage of low area overhead by using the delay measurement method.

5.5 Conclusion

The limitations on the clock frequencies and the accuracy of external testers are especially critical for testing high speed interface timing parameters. A programmable on-chip dual-capture signal generator is presented to solve the issues. A technique which generates the arbitrary capture signal (regardless of the external test clock) is used for the I/O timing tests. The technique measures the delay difference between data and clock paths instead of using complicated test vectors. The delays of '20 ps' to '0.875 tCK_{vco} + 180 ps' can be measured with the 20 ps resolution. The tCK-based delay range enables the timing tests in the wide operating range. The feature also increases the testability for the frequency-based parameters and long-delay paths. The test results obtained by the delay measurements and the comparisons decide the timing pass or fail decision, and thus find the timing margin.

Chapter 6

On-Chip Small Timing Error Detection for High-Speed Parallel I/O Timing Tests

6.1 Introduction

The delay-line based timing measurement circuits have a capability to measure the time delay with high resolution. The capability of measuring the minimum delay is, however, constrained by an offset delay due to the internal capacitances of the measurement circuits. On the other hand, with the increasing data rates, the timing specifications become tighter and the delay mismatches may be in a sub-nanosecond or sub-picosecond range, and hence we need to detect the small timing errors. We could remove the side effect due to the offset delay by adding dummy delays to compensate the offset delay between data and clock paths. However, the additional circuitry increase the PVT variations, which increase timing variations and decrease test accuracy. Moreover, this method requires a longer test time especially for testing the tCK-based timing parameters. To avoid these issues, this chapter presents a method measuring the path delay difference between two paths instead of measuring the time delay between data and strobe at the destination. Regardless of the system operating frequency, the time delay between data and strobe paths is maintained. Thus, we do not need to generate a wide delay range to measure the path delay and also this method keeps a steady test time for testing the tCK-based timing parameters.

The data and strobe signals are transferring in parallel for the source synchronous memory interfaces, and thus the path delay is very small if there are no critical design errors. To detect the path delay difference, we first measure the time delays between data and strobe at the source and the destination, respectively, and then compare the time delays to detect the path delay mismatch. Another issue is that all the I/O circuits are distributed over the chip and so it is hard to detect the path delay. Therefore, this chapter presents a digital assisted path delay mismatch detector [35]. This scheme converts the time delay to a pulse and then the I/O timing parameters are digitally calculated using a time-to-voltage converter and analog-to-digital converter (ADC). The ADC is necessary to develop a complete system-level test and obtain a low-cost test method. To decrease the area overhead of ADC, we present a novel calibration method to control the input range of ADC.

6.2 Design Methodology

In source-synchronous interfaces, the data and clock paths are designed to match. Thus, the delay mismatch between DQ and DQS is ideally zero. However, unlike the DQ path, the DQS path has a tree structure for clocking each DQ. Due to the different capacitive loadings on DQ and DQS path, there is a delay mismatch. Moreover, factors like process variations, crosstalk, Inter-Symbol Interference (ISI), and power noise increase the delay variation. The amount of mismatch between two paths directly affects the interface timing parameters. Our goal is to measure the delay difference between data and clock paths directly instead of sweeping the data-strobe with respect to the data. The technique achieves high test coverage without having to generate many data patterns required for strobe-scanning method. It also has a shorter test time compared with the strobe-scanning method. Additionally, it does not require a high performance ATE or circuits with large area overhead for phase generation and alignment.

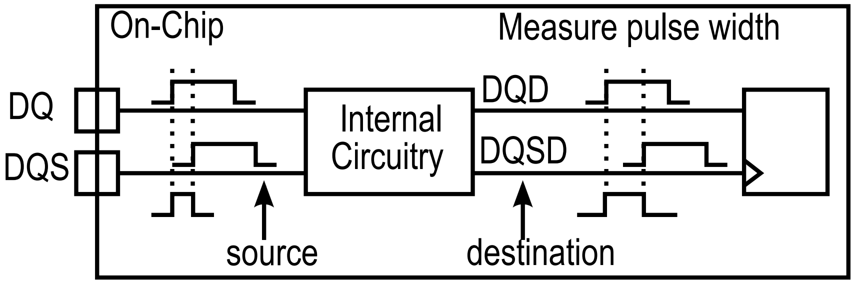


Figure 6.1: Basic idea for path delay mismatch detection

Figure 6.1 explains the basic idea of this delay mismatch detection method. While DQ and DQS are transferring from the source to the destination in the internal interface circuitry, a delay mismatch between DQ and DQS occurs. We generate a pulse with a certain pulse width at the source, and measure the pulse width variation at the destination occurred by the delay mismatches between DQ and DQS. The strength of this method is to have a steady test time regardless of the operating clock frequency. In the previous methods [32, 33], in case of the tCK-based timing parameter, tDQSS, we need to implement a wide delay range. On the other hand, this method does not require a wide delay range because we detect only the delay difference between DQ and DQS or DQS and CLK.

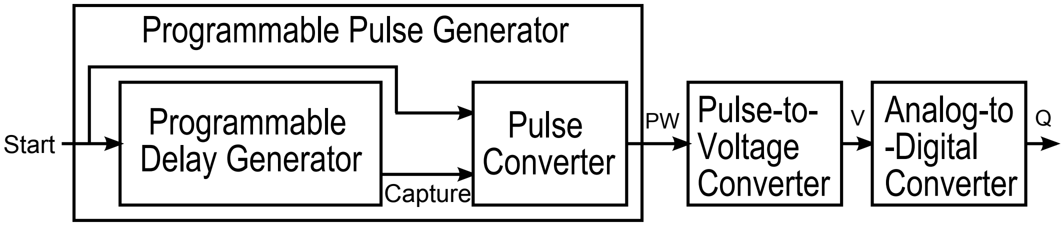


Figure 6.2: Delay mismatch detection method

Figure 6.2 shows the delay mismatch detector to generate delay and convert it into digital signal compatible with low cost ATE. Using the proposed scheme, we measure each delay difference at the source and the destination of data and clock paths to detect the delay mismatch. The scheme consists of the programmable pulse generator (PPG), the pulse-to-voltage converter (PVC), and finally an analog-to-digital converter (ADC). The programmable pulse generator (PPG) combines the programmable capture generator (PCG) and the pulse converter. We use a PCG similar to [62]. However, unlike [62], we insert pull up transistors for testability of the rising and the falling transitions. Also, this PCG does not need to be controlled precisely. The PCG generates a programmable timing interval between data and clock by adjusting the output of the PCG. We then convert the delay difference to pulse width using pulse converter. As shown in Figure 6.2, the PCG is triggered by the 'Start' signal, and the 'Capture' signal is generated with the programmable delay with respect to the 'Start'. The delay difference between 'Start' and 'Capture' is converted to a pulse using the pulse converter. The pulse width is thus controlled by the programmable capture time. The pulse width is converted to a voltage using the PVC, and ADC is used to obtain digital value of voltage. This makes this method compatible with low-cost ATE. A good resolution ADC can have large area overhead. We present a novel calibration method for input range of ADC to reduce the area overhead. Calibration of ADC makes the conversion range narrower. Since the programmable range of the pulse width is determined by the given timing specifications, the ADC input range is known. So, we adaptively calibrate the input range according to the reference timing specification.

6.3 Digital Assisted Path Delay Mismatch Detector

6.3.1 Circuit Structure for the Interface Timing Tests

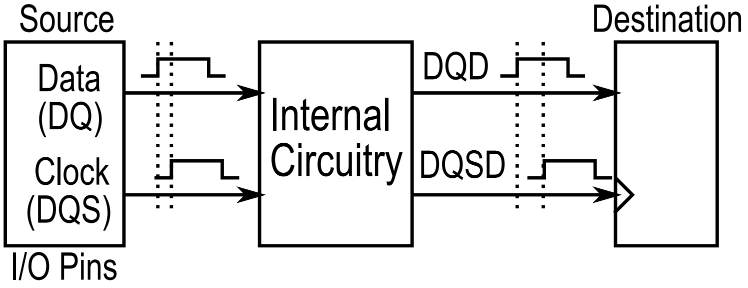


Figure 6.3: Circuit under test for delay mismatch detection

We model a circuit under test (CUT) shown in Figure 6.3 to apply the on-chip delay mismatch detector to test the interface timing parameters. The DQ and DQS are the source signals, and DQD and DQSD are the destination signals. The CUT has models of DQ and DQS paths to test the data setup and hold times in the DDR memory device. The two paths are designed to have the same delay from the I/O pads to the data register. The amount of mismatch from the source to the destination needs to be measured to check if the timing specification between DQ and DQS is satisfied. For the measurement, we compare each delay difference between DQ and DQS paths at the source and the destination by incorporating our scheme into the CUT unit. The overall on-chip path delay mismatch detector design is shown in Figure 6.4. During the test, ATE generates the input signal DQ. DQ is the 'Start' signal of the PCG. DQS is the 'Capture' signal generated in the PCG. Therefore, the PCG used at the source produces the programmable delay between DQ and DQS. Thus, the PPG at the source generates the programmable pulse width that indicates the delay between DQ and DQS. The delay difference at the source is controlled by the programmable delay. The DQ and DQS are propagated to the destination data register. The delay variation between DQD and DQSD is converted to a pulse using the pulse converter. The difference between each pulse width at the source and the destination indicates the amount of mismatch. We convert the pulse width to the corresponding voltage level which are digitally read out using ADC. This makes our design compatible with low cost ATE. We also

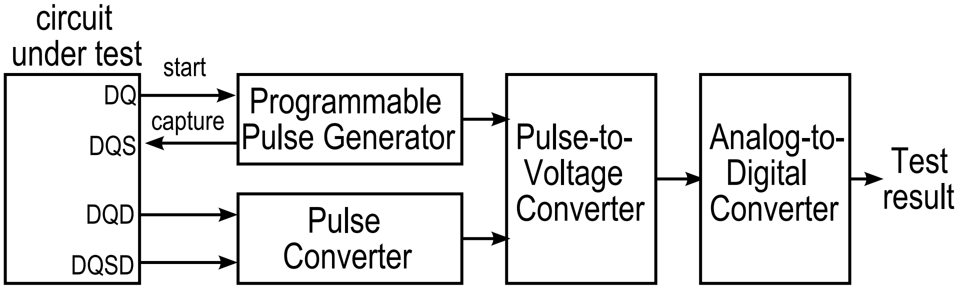
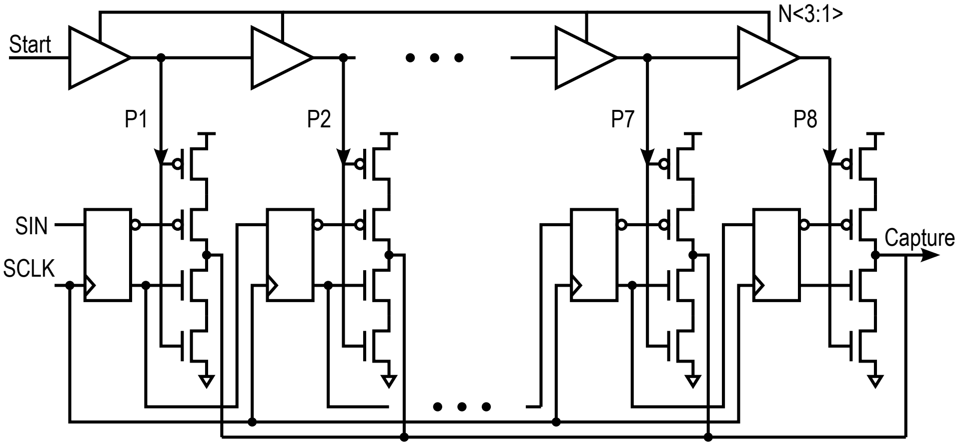


Figure 6.4: Overall structure for on-chip path delay mismatch detector

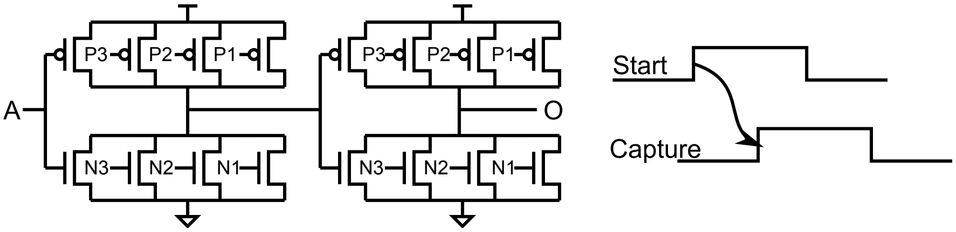
measure the timing margin by comparing the amount of mismatch from the given timing specification. Using the PCG, the delay between DQ and DQS is set to the timing specification defined in the data sheet such as [1]. The generated pulse width at source is the reference value which is compared with the pulse width at destination. The difference gives the timing margin between DQ and DQS. A novel input rage calibration method for ADC is presented to keep the ADC cost low. The detailed descriptions of all sub-circuits and algorithm are presented in following subsections.

6.3.2 Programmable Pulse Generator

As shown in Figure 6.2, the PPG consists of the PCG and the pulse converter. The schematic of the PCG is shown in Figure 6.5. The PCG generates the ‘Capture’ with the programmable delay with respect to the ‘Start’. The programmable delay is set by the control signals of the unit delay (UD) and the scannable flip-flops. The number of the unit delay selected is determined by one-hot sequence ‘SIN’. The unit delay is implemented using a segmented-type driver [14] as shown in Figure 6.5(b). Thus, it is more precisely controlled by the codes of *N1*, *N2*, and *N3*. The codes of *P1*, *P2*, and *P3* are the inverted signals of the *N1*, *N2*, and *N3*. Thus, the two-level delay control is designed with lesser area overhead compared to the PCG [62]. The delay between ‘Start’ and ‘Capture’ is kept arbitrary or pre-defined depending on the test purpose. If the goal is to measure the amount of mismatch between data and clock paths, the delay can be arbitrary. On the other hand, in the case of measuring the



(a) Schematic



(b) Unit delay (UD)

Figure 6.5: Programmable delay generator

timing margin, the delay is determined by the given timing specifications. In general, because of the combinational logic delays between the source and the destination of the interface signals, we need to compare the signals generated at different timings to detect the timing mismatch. Therefore, we convert the delays to pulses to compare the delay differences produced at different timings. The pulse converter is implemented as shown in Figure 6.6. It consists of the rising edge detectors and the SR latch. The rising edge detector produces a signal with the pulse width of three inverter delays. The SR latch simply forms the pulse width corresponding to the delay between ‘Start’ and ‘Capture’. On

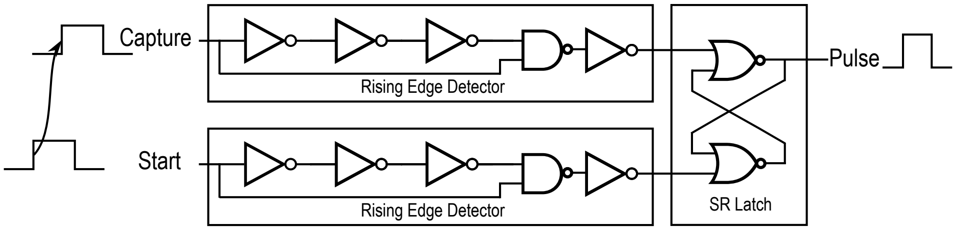
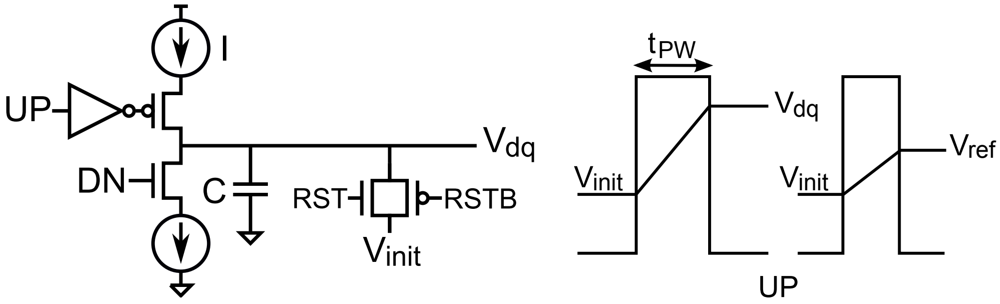


Figure 6.6: Pulse converter

the other hand, the propagation delays of the 'Set' and 'Reset' operations of the SR latch are not the same, and thus the generated pulse width is less than the actual delay between 'Start' and 'Capture'. The non-symmetrical characteristics of the SR latch is not a problem for the test circuit. This is because we measure the relative delay difference at the source and the destination by using the same pulse converter at both ends. As a result, the PPG combining the PCG and the pulse converter is used at the source and only the pulse converter is used at the destination. The pulse widths formed by the first and the second pulse converters are compared to detect the amount of mismatch or the timing margin.

6.3.3 Pulse-to-Voltage Converter

The pulse widths generated at the source and the destination are converted to voltage levels to measure the pulse difference. For this operation, the PVC is implemented as shown in Figure 6.7(a). It consists of the current steering logic and the capacitor. The generated pulse, UP, increases the voltage across the capacitor after the pulse switches on. Thus, the output voltage level, V_{dq}, is changed according to the pulse width as shown in Figure 6.7(b).



(a) Schematic (b) Operation

Figure 6.7: Pulse-to-voltage converter

As the pulse width becomes narrower, the output voltage level becomes lower. We also define V_{init} as the initial voltage level of V_{dq} to have the same initial level at the beginning of each test. Moreover, the voltage-level range can be calculated since we can estimate the range of UP pulse. Thus, C, the capacitor value, is also determined by the Equation (6.1) during the design. The capacitor needs to be fully charged to V_{dq} during the pulse period. Accordingly, the capacitor value is determined by the pulse width of UP and the maximum variation between V_{init} and V_{dq}. Therefore, we need to consider the maximum allowable pulse width during the design to select the optimum capacitor value.

$$\mathrm{C} = \frac{\Delta \mathrm{t}_{\mathrm{PW}_{\mathrm{max}}}}{\Delta \mathrm{V}_{\mathrm{max}}}, \quad \Delta \mathrm{V}_{\mathrm{max}} = \Delta \mathrm{t}_{\mathrm{PW}_{\mathrm{max}}}/\mathrm{C} \tag{6.1}$$

$$\mathrm{V}_{\mathrm{dq,\ max}} = \mathrm{V}_{\mathrm{init}} + \Delta \mathrm{V}_{\mathrm{max}}$$

Figure 6.8 shows the timing diagram for the circuit operations. The V_{ref} is the reference value generated by the given timing specification. V_s is generated using DQ and DQS signals at the source. V_d is produced using DQD and DQSD signals at the destination. Each voltage level is compared with V_{ref},

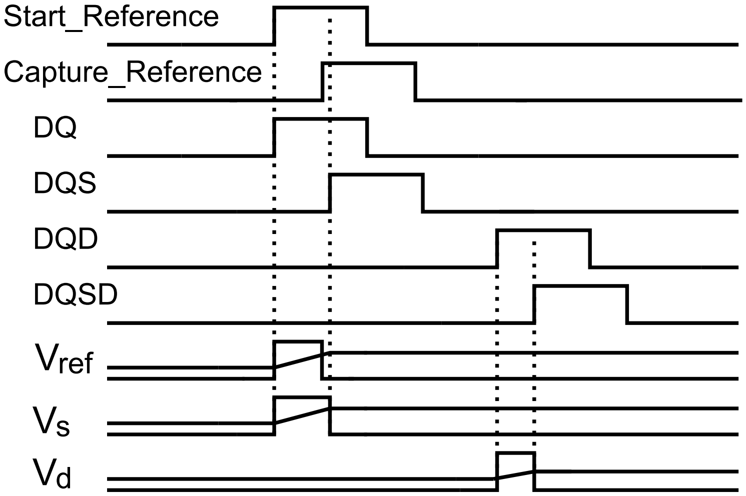


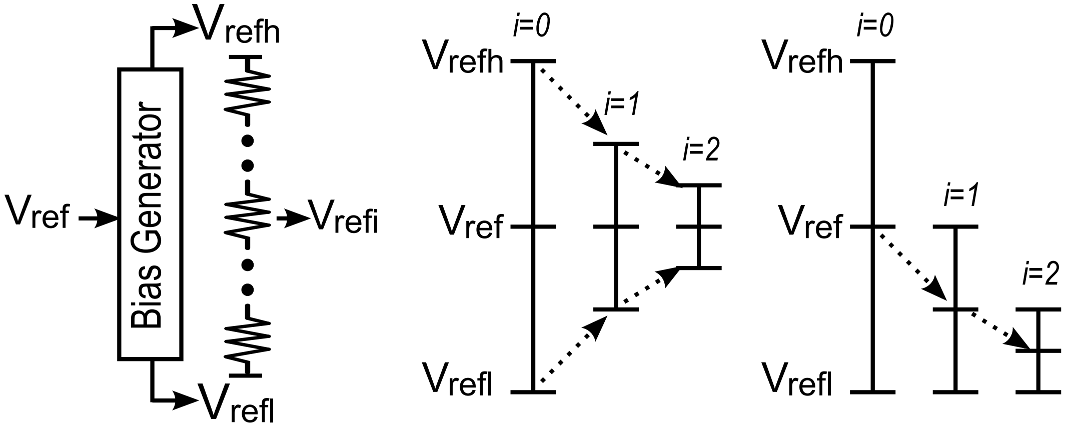
Figure 6.8: Timing Diagram

and their differences are measured using the ADC.

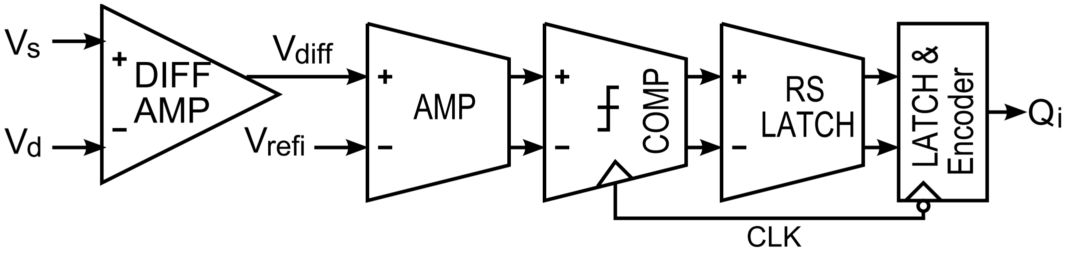
6.3.4 Analog-to-Digital Converter with Calibration

The ADC is used to read out the test results digitally which makes it compatible with low cost ATE. The ADC structure is shown in Figure 6.9(b). We use a flash ADC because of its simple structure. The flash ADC consists of a bias generator, a resistor ladder, differential amplifiers, comparators, latches, and an encoder [7]. However, it consumes large area and power to achieve high resolution because it needs $2^N - 1$ comparators and amplifiers, respectively, where N is the ADC resolution.

We use a novel calibration method to reduce the number of ADC bits without compromising high resolution. The goal of this calibration method is to adaptively change the input range (difference between V_{refl} and V_{refh}) of



(a) VREF calibration



(b) Analog-to-digital converter

Figure 6.9: Analog-to-digital converter with calibration

ADC. V_{diff} indicates the pulse width difference between source and destination. Since input pulse width (V_{diff}) is known, we use it to calibrate the input range of ADC. Figure 6.9(a) shows two ways of calibrating the input range of ADC. One is to change V_{refl} and V_{refh} and the other is to change V_{ref}. V_{ref} is the generated signal at source based on the given timing specification. The ADC reference voltages of V_{refl}, V_{refh} and V_{ref} are defined by the following Equation (6.2), where V_{ref} and the ΔV_{max} are known.

$$\mathrm{V_{refh}} = \mathrm{V_{ref}} + \mathrm{V_m}\ ,\ \mathrm{V_{refl}} = \mathrm{V_{ref}}\ \text{-}\ \mathrm{V_m}\ ,\ \mathrm{V_m} = \frac{1}{2}\Delta \mathrm{V_{max}} \tag{6.2}$$

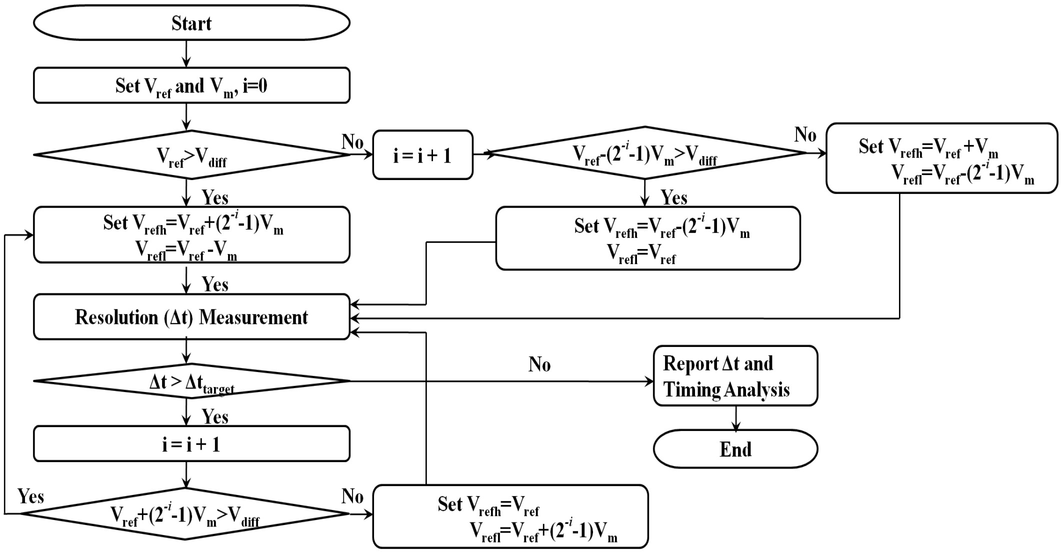


Figure 6.10: Calibration flow chart

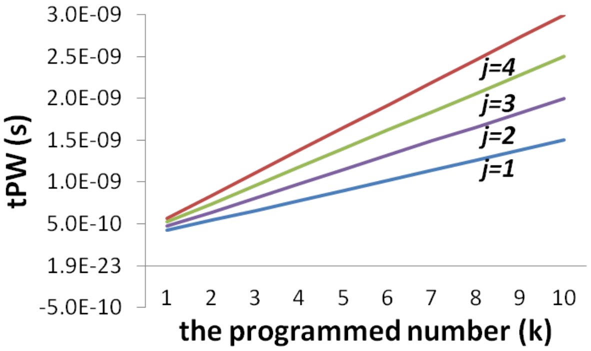
The calibration flow chart for the V_{refl} and the V_{refh} is described in Figure 6.10. V_{diff} lower than the V_{ref} means that the timing specifications are satisfied. V_{refl} and V_{refh} are initially set to '$V_{ref}\pm V_m$'. The calibration uses the binary search algorithm which compares V_{diff} with the middle of the reference levels. This process minimizes the difference between the reference levels and V_{diff}. At initial step, *i=0*, the test circuit determines the pass or fail for the timing specifications by comparing the V_{diff} with the V_{ref}. After the decision, the test resolution is calculated and then compared with the target resolution, Δt_{target}. If the test resolution is not within the tolerance range compared to the target value, the reference levels are narrowed down. These steps are repeated until the test resolution is within the tolerance range. Because of the binary search algorithm, we obtain the target resolution with only a few calibration steps. After the calibration is completed, the test circuit reports the test resolution and the ADC outputs, and then analyze the timing parameters. Thus, we obtain the high test resolution using a small number of ADC bits.

6.4 Simulation Results

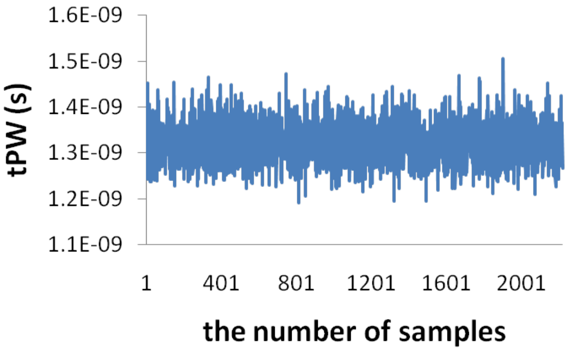
The on-chip test circuit shown in Figure 6.4 is implemented using 0.18 μm CMOS process. Transistor-level simulations using Hspice as well as behavior-level simulations using Matlab are performed to validate the test circuit and the calibration technique, respectively. Figure 6.11(a) shows the programmed pulse-width, the simulation result of the PPG circuit. According to the number of the selected unit delay, the pulse width is represented by the following Equation (6.3),

$$\begin{aligned} t_{\mathrm{PW}} &= t_{\mathrm{uk}} + t_{\mathrm{offset}} \\ t_{\mathrm{uk}} = k \times (t_{\mathrm{c}}+ t_{\mathrm{fj}});\ t_{\mathrm{fj}} &= j \times t_{\mathrm{f}}; \\ k = [1,\ N_c],\ j &= [1,\ N_{\mathrm{f}}] \end{aligned} \tag{6.3}$$

where t_{uk} is the programmed delay and t_{offset} is the constant offset delay due to the capacitance loadings. t_c is the one unit delay and t_f is one segment unit delay where N_c is the number of unit delay and N_f is the number of segmented unit delay. For the PCG, 10 unit delay are used and each unit delay is segmented into 3 units. According to the programmed numbers, k and j, the programmed pulse width is shown in Figure 6.11(a) for a range from 500 ps to 3 ns. Figure 6.11(b) shows the result of 2000-trial Monte Carlo simulations



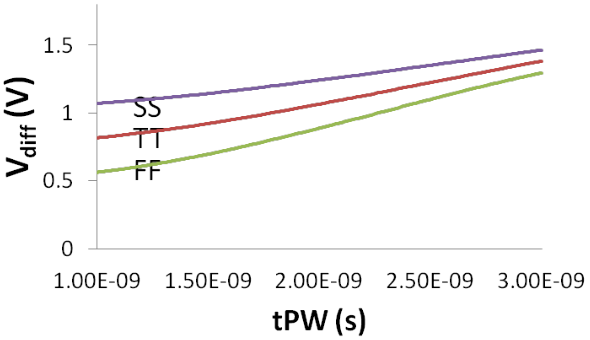
(a) The programmed pulse-width



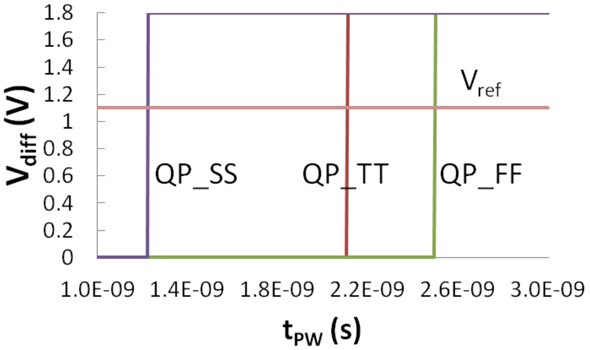
(b) Pulse-width variation

Figure 6.11: Programmable pulse generator

to estimate the pulse-width variations due to intra-chip parameter variations. The 200 ps variation for the input range of ADC is injected for its calibration. Since we know the pulse width and its variation as shown in Figure 6.11, our calibration technique is effectively applied. The pulse width is converted to a voltage level to measure the test results. To validate the conversion relationship, simulations for the PVC scheme are performed. Figure 6.12(a) shows the relationship between t_{PW} and V_{diff} for different process corners. As



(a) Pulse width vs. voltage level



(b) Timing failure decision

Figure 6.12: Pulse-to-voltage converter

can be seen, V_{diff} is changed from 0.8 V to 1.36 V while t_{PW} is controlled from 1 ns to 3 ns under the typical conditions. In other words, the voltage level variation is 0.56 V while the pulse width variation is 2 ns. Accordingly, the conversion ratio, r, between two parameters is approximately '*2.8 mV/10 ps*'. Since the the pulse width is linearly converted to the voltage level, the parameterizations of the test results are available. After the conversion, V_{ref} is chosen to start the calibration. At the first calibration step, the timing pass or fail is determined by comparing V_{diff} with V_{ref}. The timings fail when V_{diff} is larger than V_{ref}. For the simulation, we set V_{ref} to 1.08 V, which is the middle value of V_{diff} range under the typical conditions. However, $\mathrm{t_{PW}}$ and V_{ref} can be changed depending on the required timing specifications. Figure 6.12(b) shows the comparison results of V_{diff} and V_{ref}, QP, to show the points where the timing failure occurs. The difference between V_{diff} and V_{ref} indicates the timing margin. We precisely find the difference between V_{diff} and V_{ref} through the calibration procedure. We extract the test resolution and then compare with the target resolution at each calibration step. The test resolutions are calculated by equation (6.4) where i indicates the calibration step.

$$\mathrm{I} \times \Delta \mathrm{t} = \mathrm{C}\Delta \mathrm{V} = \mathrm{C}\frac{\Delta V_{\mathrm{max}}}{2^{\mathrm{i}}}\frac{\mathrm{I}}{2^{\mathrm{N}}} \tag{6.4}$$

The resolutions are calculated as shown in Table 6.1. The ΔV_{max} is changed by the variable reference levels during the calibration procedure. By using

i	ΔV_{max}	Δt
1	0.8 V	88 ps
2	0.8 V	44 ps
3	0.8 V	22 ps
1	0.4 V	44 ps
2	0.4 V	22 ps
3	0.4 V	11 ps
1	0.1 V	11 ps
2	0.1 V	6 ps
3	0.1 V	3 ps

Table 6.1: Resolution calculation

the binary search algorithm, we obtain sufficient resolution only at the third step of calibration. Therefore, only a 4-bit flash ADC is required to measure the test results with high resolution. After the calibration, the ADC digital outputs, Q_i, are reported and the timing margin, $\mathrm{t_{MD}}$, is parameterized using the $\mathrm{t_{PW}}$, Δt, and the conversion ratio as shown in the equation (6.5).

$$\mathrm{t_{MD}} = |t_{\mathrm{PW}}(V_{\mathrm{ref}}) \text{ - } t_{\mathrm{PW}}(V_{\mathrm{diff}})| \times \mathrm{r} \times \Delta\mathrm{t} \times Q_{\mathrm{i}} \tag{6.5}$$

Using the 0.18 μm CMOS process, the chip size is '0.14 mm × 0.12 mm' as

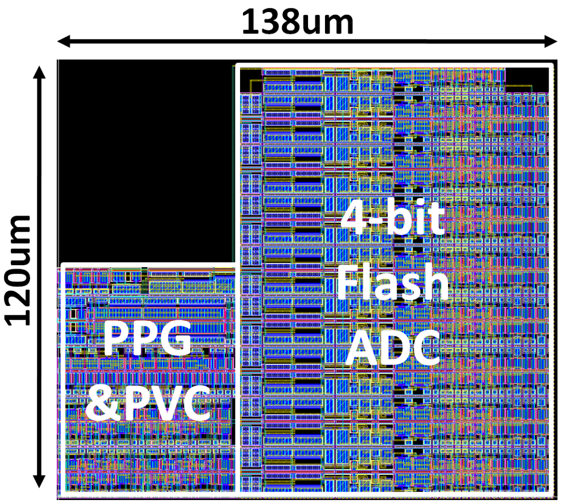


Figure 6.13: Layout

shown in Figure 6.13. Hence, we achieved low area overhead while having high resolution.

6.5 Conclusion

This chapter has addressed the issues of increasing test cost and time for testing high speed source synchronous interface timings. To reduce test time, we present a method to measure the timing parameters by detecting a small path delay mismatch. The test methodology is based on the delay measurement of data and clock paths instead of the strobe-scanning method. In order to measure the memory I/O timing parameters, we detect the small path delay mismatches between data and clock paths by comparing the time delays at the start and the end points of two signal paths. The path delay mismatches are calculated to obtain the I/O timing specifications. The test results are converted to digital values using ADC, and a novel calibration technique is presented for a low hardware overhead. Because of the predictable ADC input range, the calibration technique is effectively applied to save both test time and hardware overhead. We have achieved around 20 ps resolution by using a 4-bit flash ADC and calibrated in only 3 steps.

Chapter 7

Conclusions

This dissertation deals with on-chip timing measurement methods to target a high test quality and low manufacturing cost for testing high-speed source synchronous DDR memory interfaces. For this goal, a built-in self-test scheme including on-chip timing samplers was implemented and fabricated in 0.18-μm CMOS technology.

The most critical timing parameters of the DDR memory interfaces, the input setup and hold time and the output timing skews, are calculated using the BIST scheme. In order to effectively test these parameters, a circuit under test was also implemented along with the on-chip measurement circuits. The BIST scheme does not require any high speed signallings between testers and DUT for testing the timing parameters of the CUT, which enables our work to be compatible with low cost testing equipments. Another on-chip measurement method, the path delay mismatch detector, was also designed and simulated to improve testability for higher speed systems. This method does not require any complicated test vectors and long delay lines because it detects a small path delay mismatch between source synchronous data and strobe paths. This scheme thus reduces test time and also keeps a steady test time regardless of the system operating frequency while the low hardware overhead is maintained.

The on-chip timing measurement techniques improve the test accuracy by avoiding external errors and enhance the test cost efficiency. Because of these strengths, the applications for our techniques will be increasing. Some of the future directions that can be explored are as follows.

- The increasing features of the future semiconductor market are high volume production and high levels of integration. Thus, the test methods for the interfacing signals severely affected by external test loads or channel noises should be discussed by both design and test viewpoints. From the beginning of design implementations, a design-for-test scheme needs to be considered to build a system-on-chip. To complete the SoC design, a common interface standard is necessary and the test circuits for the interfaces should also be able to characterize a variety of interface standards.
- With the increasing high data rates, test costs take a larger portion of IC production costs. However, lower performance testers cause more timing uncertainties and decrease test accuracy due to low-quality test signals. Thus, it is important to implement a design-for-test scheme to compensate for the low performance of ATE. We can thus extend our scheme for timing mismatch decompositions that distinguish off-chip

and on-chip timing mismatches. This work can be done by distributing properly the on-chip measurement circuits near critical signal paths.

- In order to decrease IC high volume production costs, a design-for-test scheme that calibrates and compensates timing errors is desired. The measurement results for the timing parameters can be used to calibrate and compensate the timing errors. In future work, a digital feedback tuning scheme can be combined with our current work, and this future work should be reviewed in the beginning process of design implementation.

Bibliography

[1] DDR2 SDRAM Specification. In *JEDEC JESD79-2C, JEDEC Solid State Technology Association*, 2006.

[2] DDR3 SDRAM Specification. In *JEDEC JESD79-3, JEDEC Solid State Technology Association*, 2008.

[3] A. Khoche, R. Kapur, D. Armstrong, T.W. Williams, M. Tegethoff, and J. Rivoir. A New Methodology for Improved Tester Utilization. In *Proc. Int. Test Conf.*, pages 916–923, 2000.

[4] A.C. Evans. Applications of Semiconductor Test Economics, and Multisite Testing to Lower Cost of Test. In *Proc. Int. Test Conf.*, pages 113–123, 1999.

[5] B. Analui, A. Rylyakov, S. Rylov, M. Meghelli, and A. Hajimiri. A 10Gb/s Eye-Opening Mnitor in 0.13um CMOS. In *ISSCC Digest of Technical Papers*, pages 332–333, 2005.

[6] B. Analui, A. Rylyakov, S. Rylov, M. Meghelli, and A. Hajimiri. A 10Gb/s Two-Dimensional Eye-Opening Mnitor in 0.13um Standard CMOS. In *IEEE J. Solid-State Circuits*, pages 2689–2699, 2005.

[7] R. Jacob Baker, Harry W. Li, and David E. Boyce. CMOS Circuit Design, Layout and Simulation. In *Wiley-IEEE Press*, 1998.

[8] I. Bayraktaroglu, O. Caty, and Y. Wong. Highly Configurable Programmable Built-In Self Test Architecture for High-Speed Memories. In *Proc. VLSI Test Sympo.*, pages 21–26, 2005.

[9] R.Z. Bhatti, M. Denneau, and J. Draper. Data Strobe Timing of DDR2 using a Statistical Random Sampling Technique. In *Proceedings of the 50th IEEE International Midwet Symposium on Circuits and Systems*, pages 1114–1117, 2007.

[10] B. Casper, A. Martin, J.E. Jaussi, J. Kennedy, and R. Mooney. An 8-Gb/s simultaneous bidirectional link with on-die waveform capture. In *IEEE J. Solid-State Circuits*, volume 38, pages 2111–2119, 2003.

[11] J. Cheng. When Zero Picoseconds Edge Placement Accuracy is Not Enough. In *Proc. Int. Test Conf.*, pages 1134–1142, 2001.

[12] J. Cheng and L. Milor. A BIST Solution for the Test of I/O Speed. In *Proc. Int. Test Conf.*, pages 1023–1030, 2003.

[13] J. Cheng and L. Milor. A DLL Design for Testing I/O Setup and Hold times. In *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, volume 17, pages 1579–1592, 2009.

[14] William J. Dally and John W. Poulton. Digital Systems Engineering. In *Cambridge University Press*, 1998.

[15] B.F. Dutton and C.E. Stroud. Built-In Self-Test of Programmable Input/Output Tiles in Virtex-5 FPGA. In *Proc. IEEE Southeastern Symp. on System Theory*, pages 235–239, 2009.

[16] M.A. El-Moursy and E.G. Friedman. Exponentially tapered H-tree clock distribution networks. In *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, volume 13, pages 971–975, 2005.

[17] T. Ellermeyer, U. Langmann, B. Wedding, and W. Pohlmann. A 10-Gb/s Eye-Opening Monitor IC for Decision-Guided Adaptation of the Frequency Response of an Optical Receiver. In *IEEE J. Solid-State Circuits*, pages 1958–1963, 2000.

[18] M.E.S. Elrabaa. A Digital Clock Re-Timing Circuit for On-Chip Source-Synchronous Serial Links. In *International Conference on Microelectronics*, pages 206–209, 2006.

[19] M. Matsui et al. A 200MHz 13 mm 2-D DCT macrocell using senseamplifying pipeline flip-flop scheme. In *IEEE J. Solid-State Circuits*, volume 29, pages 1482–1490, 1994.

[20] J.A. Gasbarro and M.A. Horowitz. Techniques For Characterizing DRAMS with a 500MHz Interface. In *Proc. Int. Test Conf.*, pages 516–525, 1994.

[21] A. Grochowski, D. Bhattacharya, T.R. Viswanathan, and K. Laker. Integrated Circuit Testing for Quality Assurance in Manufacturinng History,

Current Status, and Future Trends. In *IEEE Transactions on Circuits and Systems*, volume 44, pages 610–633, 1997.

[22] H. Vranken, T. Waayers, H. Fleury, and D. Lelouvier. Enhanced Reduced Pin-Count Test for Full-Scan Designs. In *Proc. Int. Test Conf.*, pages 738–747, 2001.

[23] http://www.jedec.org, JEDEC Solid State Technology Association.

[24] C.C. Huang, K.S. Oh, and S. Rajan. The Interconnect Design and Analysis of RAMBUS Memory Channel. In *Proceedings of ASME IPACK2001, IPACK2001-15531*, 2001.

[25] J.-H. Kim, W. Kim, D. Oh, R. Schmitt, J. Feng, C. Yuan, L. Luo, and J. Wilson. Performance Impact of Simultaneous Switching Output Noise on Graphic Memory Systems. In *In Proc.16th Topical Meeting on Elect. Perform. Electron. Packag.*, pages 197–200, 2007.

[26] Y.-C. Jang, J.-Y. Park, S.C. Shin, H.D. Choi, K.S. Lee, B.S. Woo, H.W. Park, W.-S. Kim, Y.D. Choi, J.K. Kim, H.-K. Kim, J.Y. Kim, S.Y Lim, S.-J. Chung, S.R. Kim, J.H. Yoo, and C.H. Kim. Self-calibrating transceiver for source synchronous clocking system with on-chip TDR and swing level control scheme. pages 54–55, 2009.

[27] J.P. Jansson, A. Mantyniemi, and J. Kostamovaara. A CMOS time-to-digital converter with better than 10ps single-shot precision. In *IEEE J. Solid-State Circuits*, volume 41, pages 1286–1296, 2006.

[28] K.A. Jenkins and L. Li. A Scalable, Digital BIST Circuit for Measurement and Compensation of Static Phase Offset. In *Proc. VLSI Test Sympo.*, pages 185–188, 2009.

[29] Howard W. Johnson and Martin Graham. High-Speed Digital Design. In *Prentice Hall*, 1993.

[30] J. Kang and G. Erickson. Testing High-Speed Protocol Based Rambus DRAM. In *Hewlett-Packard Company California Semiconductor Test Division Application Note*, 1999.

[31] D.C. Keezer, D. Minier, and P. Ducharme. Source-Synchronous Testing of Multilane PCI Express and HyperTransport Buses. In *IEEE Design & Test of Computers*, volume 23, pages 46–57, 2006.

[32] H.J. Kim and J.A. Abraham. A Low Cost Built-In Self-Test Circuit For High-Speed Source Synchrounous Memory Interfaces. In *Proc. Asian Test Sympo.*, pages 123–128, 2010.

[33] H.J. Kim and J.A. Abraham. On-Chip Programmable Dual-Capture for Double Data Rate Interface Timing Test. In *Proc. Asian Test Sympo.*, pages 15–20, 2011.

[34] H.J. Kim and J.A. Abraham. A BIST Solution for DDR Memory Output Timing Test and Measurement. In *Proc. VLSI Test Sympo.*, 2012.

[35] H.J. Kim and J.A. Abraham. On-Chip Source Synchronous Interface Timing Test Scheme with Calibration. In *Proc. Design, Automation and Test in Europe*, 2012.

[36] H.J. Kim, J.Y. Chung, J.A. Abraham, E.J. Byun, and C.-J. Woo. A Built-In Self-Test Scheme for High Speed I/O Using Cycle-by-cycle Edge Control. In *Proc. European Test Sympo.*, pages 145–150.

[37] K. Kim, J. Hwang, Y.B. Kim, and F. Lombardi. Data Dependent Jitter (DDJ) Characterization Methodology. In *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 294–302, 2005.

[38] B. Laquai, M. Braun, S. Walther, and G. Schulze. Flexible and scalable methodology for testing high-speed source synchronous interfaces on automated test equipment (ATE) with multiple fixed phase capture and compare. In *IET Computers & Digital Techniques*, volume 1, pages 154–158, 2007.

[39] J.-B. Lee, K.-H. Kim, C. Yoo, S. Lee, O.-G. Na, C.-Y. Lee, H.-Y. Song, J.-S. Lee, Z.-H. Lee, K.-W. Yeom, H.-J. Chung, I.-W. Seo, M.-S. Chae, Y.-H. Choi, and S.-I. Cho. Digitally-controlled DLL and I/O circuits for 500Mb/s/pin X 16 DDR SDRAM. In *ISSCC Digest of Technical Papers*, pages 68–69, 2001.

[40] M. Li. Requirements, Challenges, And Solutions For Testing Multiple GB/s ICs In Production. In *Proc. Int. Test Conf.*, page 1309, 2003.

[41] M.P. Li. Jitter Challenges and Reduction Techniques at 10Gb/s and Beyond. In *IEEE Trans. on Advanced Packaging*, volume 32, pages 290–297, 2009.

[42] M. Suda, K. Yamamoto, T. Okayasu, S. Kantake, S. Sudou, and D. Watanabe. CMOS High-Speed, High-Precision Timing Generator for 4.266-Gbps Memory Test System. In *Proc. Int. Test Conf.*, pages 858–866, 2005.

[43] N.R. Mahapatra and B. Venkatrao. The Processor-Memory bottleneck: Problems and Solutions. In *ACM Crossroad*, 1999.

[44] T.M. Mak, M. Tripp, and A. Meixner. Testing Gbps interfaces without a gigahertz tester. In *IEEE Design & Test of Computers*, pages 278–286, 2004.

[45] W.R. Mann, F.L. Taber, P.W. Seitzer, and J.J. Broz. The Leading Edge of Production Wafer Probe Test Technology. In *Proc. Int. Test Conf.*, pages 1168–1195, 2004.

[46] H. Muljono, B.-T. Lee, Y. Tian, Y. Wang, M. Atha, T. Huang, M. Adachi, and S. Rusu. A 400-MT/s 6.4-GB/s multiprocessor bus interface. In *IEEE J. Solid-State Circuits*, volume 38, pages 1846–1856, 2003.

[47] O. Weeden. http://www.keithley.com/, Probe Card Tutorial. 2003.

[48] D. Oh, W. Kim, B. Stott, L. Yang, and C. Yuan. Channel timing error analysis for ddr2 memory system. In *IEEE 14th Topical Meeting on Electrical Performance of Electronic Packaging*, pages 119–122, 2005.

[49] P. Dudek, S. Szczepanski, and J. Hatfield. A High Resolution CMOS Timing-to-Digital Converter Utilizing a Vernier Delay line. In *IEEE J. Solid-State Circuits*, volume 35, pages 240–247, 2000.

[50] David A. Patterson and John L. Hennessy. Computer Organization and Design. In *Morgan Kaufmann Publishers*, 1997.

[51] B. Provost, T. Huang, C.H. Lim, K. Tian, M. Bashir, M. Atha, A. Muhtaroglu, C. Zhao, and H. Muljono. AC IO loopback design for high speed uProcessor IO test. In *Proc. Int. Test Conf.*, pages 23–30, 2004.

[52] R. Datta, A. Sebastine, and J.A. Abraham. Delay Fault Testing and Silicon Debug Using Scan Chains. In *Proc. European Test Sympo.*, pages 23–26, 2004.

[53] R. Nair, S.M. Thatte and J.A. Abraham. Efficient Algorithms for Testing Semiconductor Random Access Memories. In *IEEE Trans. on Computers*, volume C-27, pages 572–576, 1978.

[54] R. Treur and V.K. Agarwal. Built-In Self-Diagnosis for Repairable Embedded RAMs. In *IEEE Design & Test of Computers*, volume 10, pages 24–33, 1993.

[55] E. Raisanen-Ruotsalainen, T. Rahkonen, and J. Kostamovaara. An integrated time-to-digital converter with 30-ps single-shot precision. In *IEEE J. Solid-State Circuits*, volume 35, pages 1507–1510, 2000.

[56] I. Robertson, G. Hetherington, and T. Leslie. Testing High-Speed, Large Scale Implementation of SerDes I/Os on Chips Used in Throughput Computing Systems. In *Proc. Int. Test Conf.*, pages 992–999, 2005.

[57] S. Sunter and A. Roy. BIST for Phase-Locked Loops in Digital Applications. In *Proc. Int. Test Conf.*, pages 532–540, 1999.

[58] S.-C. Shen, H.-M. Hsu, Y.-W. Chang, and K.-J. Lee. A High Speed BIST Architecture for DDR-SDRAM Testing. In *Proc. IEEE Int. Workshop on Memory Technology Design and Testing*, pages 52–57, 2005.

[59] S. Sidiropoulos, D. Liu, J. Kim, G.-Y. Wei, and M. Horowitz. Adaptive bandwidth DLLs and PLLs using regulated supply CMOS buffers. In *Proc. IEEE Annual Symp. on VLSI*, pages 124–127, 2000.

[60] A.T. Sivaram, M. Shimanouchi, H. Maassen, and R. Jackson. Tester Architecture For The Source Synchronous Bus. In *Proc. Int. Test Conf.*, pages 738–747, 2004.

[61] S.K. Sunter and B. Nadeau-Dostie. Complete, contactless I/O testing - Reaching the boundary in minimizing digital IC testing cost. In *Proc. Int. Test Conf.*, pages 446–455, 2002.

[62] R. Tayade and J.A. Abraham. On-chip Programmable Capture for Accurate Path Delay Test and Characterization. In *Proc. Int. Test Conf.*, pages 1–10, 2008.

[63] The International Technology Roadmap for Semiconductors (ITRS) 2009: http://public.itrs.net/.

[64] D. Topisirovic. Advances in VLSI Testing at MultiGb per Second Rates. In *Serbian Journal of Electrical Engineering*, volume 2, pages 43–55, 2005.

[65] M. Tripp, T.M. Mak, and A. Meixner. Elimination of traditional functional testing of interface timings at Intel. In *Proc. Int. Test Conf.*, pages 1014–1022, 2003.

[66] E.H. Volkerink, A. Khoche, J. Rivoir, and K.D. Hilliges. Test Economics for Multi-site Test with Modern Cost Reduction Techniques. In *Proc. VLSI Test Sympo.*, pages 411–416, 2002.

[67] J. Vollrath, J. Schwizer, M. Gnat, R. Schneider, and B. Johnson. DDR2 DRAM Output Timing Optimization. In *Proc. Int. Test Conf.*, pages 858–866, 2005.

[68] S. Wang and L. Wang. Analysis of deskew signaling via adaptive timing. In *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, volume 28, pages 601–605, 2009.

[69] D. Watanabe, M. Suda, and T. Okayasu. 34.1Gbps low jitter, low BER high-speed parallel CMOS interface for interconnections in high-speed memory test system. In *Proc. Int. Test Conf.*, pages 1255–1262, 2004.

[70] K. Yamamoto, M. Suda, and T. Okayasu. 2GS/s, 10ps Resolution CMOS Differential Time-to-Digital Converter for Real-Time Testing of Source-Synchronous Memory Device. In *Proc. IEEE Custom Integrated Circuits Conf.*, pages 145–148, 2007.

[71] K. Yamamoto, M. Suda, T. Okayasu, H. Niijima, and K. Tanaka. Multi Strobe Circuit for 2.133GHz Memory Test System. In *Proc. Int. Test Conf.*, pages 1–9, 2006.

[72] R. Yerganian. Vitesse Purchases Fusion for Testing SERDES, Serial-ATA and PCI Express Devices. In *Business Wire*, 2004.

Vita

Hyunjin Kim was born in Junju, Republic of Korea in 1977. She received the Bachelor of Science degree in Electrical Engineering from Ewha Womans University in 1999. She then received the Master of Science degree in Electrical Engineering from Korea Advanced Institute of Science and Technology in 2001. In 2001, she started to work as an engineer and participated to develop Very-high-bit-rate Digital Subscriber Line chipset and SerDes chip for a startup design company in Seoul, Korea. She left the design company in 2003 and joined Samsung Electronics. She worked as a senior engineer on high-speed memory I/O and PLL/DLL circuit designs for Samsung Electronics before she started her Ph.D. work in 2008. In 2008, she joined the University of Texas at Austin, where she received the Master of Science degree in Electrical and Computer Engineering in 2011 and continued to pursue her Ph.D. degree on Mixed signal design and test.

Permanent address: eledrmac@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.